P. 1877/74

11 1974

informatyka

SPIS TREŚCI

str.

O Studium Informatyki UW. Rozmowa z prof. dr STANISŁAWEM TURSKIM — *Krystyn Bernatowicz* 1

ZSRR: postępy automatyzacji zarządzania — *Dorota Prawdzic* 5

Główne kierunki badawcze w dziedzinie oprogramowania — *Władysław M. Turski* 9

Problemy integracji systemu informacyjnego — *Zygmunt Ryznar* 13

Język programowania PASCAL i jego kompilacja — *Stanisław Matwin* 18

Aktualne tendencje rozwoju sprzętu i oprogramowania — tłum. i oprac. *J. Kiełbasiński i J. Sobczyk* 22

Problemy socjologiczne systemów komputerowych tłum. i oprac. *M. Bojańczyk* 32

Z KRAJOWEGO BIURA INFORMATYKI I ZJEDNOCZENIA INFORMATYKI JS EMC

System obrotu lekami — *J. Czubek* 25

Oprogramowanie Jednolitego Systemu. System kartotek bazowych — *J. Grzemowski* 34

Z KRAJU

Informatyka w liceach — *B. Stachura* 37

Ośrodek Obliczeniowy Akademii Rolniczej w Poznaniu — *H. Hudzik, St. Czajka* 39

Kartka z Kielc 40

ZE ŚWIATA

IFIP 74 — *W. Klepacz* 41

Rozwój współpracy w Jednolitym Systemie 44

Międzynarodowe kursy EPD w Budapeszcie 45

OŚRODKI INFORMATYKI PREZENTUJĄ

Praca ODRY 1305 w systemie operacyjnym GEORGE-2 — *K. Marks* 46

PROBLEMATYKA BAZY DANYCH

System zarządzania bazą danych SYKON — *Władysław Bogucki, Witold Staniszkis* 47

PRZEGLĄD WYDAWNICTW

Prawo — jego model cybernetyczny — *Jan Szymański* 51

Ogłoszenia III i IV okł.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

mgr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), doc. dr inż. Zbigniew GACKOWSKI, mgr inż. Marek HOŁYŃSKI, Władysław KLEPACZ, Elżbieta KOŁODZIEJSKA, doc. dr hab. Antoni MAZURKIEWICZ, inż Dorota PRAWDZIC (zast. redaktora naczelnego)

Sekretarz Redakcji mgr Krystyna WROŃSKA **Red. tech. Józef DUSZA**

RADA PROGRAMOWA

Przewodniczący — Prof. dr hab. Andrzej STRASZAK

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 28-82-61 w. 285, dyżury redakcji 10,00—13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka. Z. 2. Zam. 520. Papier druk. sat. IV kl. 70 g 61×86. Obj. 4 ark. druk. Nakład 6200+30. W-50.

Cena egzemplarza zł 8.— **INDEKS 36707** **Prenumerata roczna zł 96.—**

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce

Nr 11

MIESIĘCZNIK

1974

ROK X

Listopad

ORGAN KRAJOWEGO BIURA INFORMATYKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

Biblioteka Techniczna

Centralny Ośrodek Badawczo-Projektowy Budownictwa Ogólnego we Wrocławiu

BIBLIOTEKA POLITECHNIKA GDAŃSKA

P. 1877/74

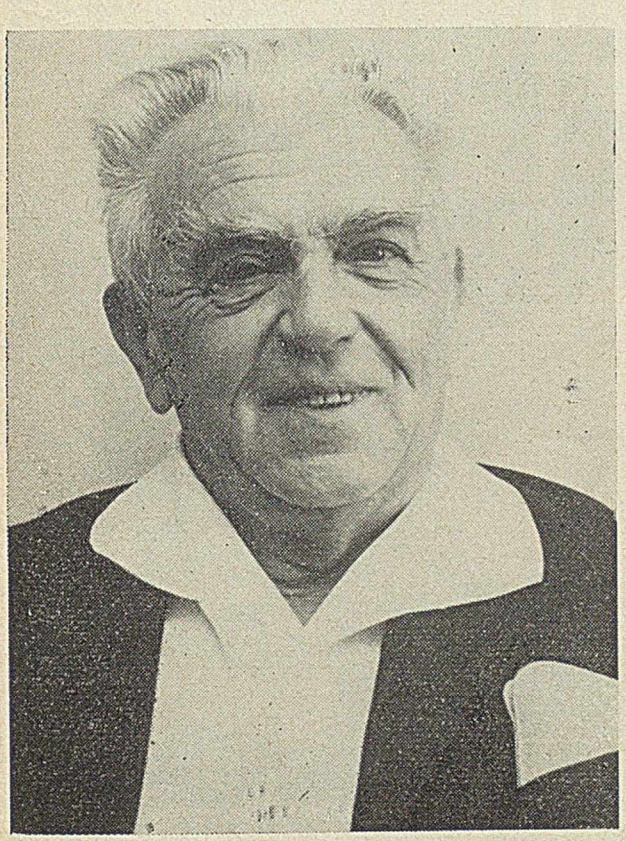
6. GRU. 1974

Rejestr czasop. Nr 744

O Studium Informatyki UW.

Rozmowa z prof. dr Stanisławem Turskim

Przed trzema laty powołano pierwsze w kraju Studium Informatyki w ramach Wydziału Matematyki i Mechaniki Uniwersytetu Warszawskiego. W październiku br. progi uczelni przekroczyła po raz czwarty ponad 100 osobowa grupa kandydatów na kierunek informatyczny. Wydaje nam się, że doświadczenia dydaktyczne Studium stanowią już poważny wkład w wiedzę o kształceniu obecnej i przyszłej kadry informatyków. Z prośbą o informację o dotychczasowym przebiegu nauczania zwróciliśmy się do kierownika Studium, prof. dr hab. Stanisława Turskiego.



Prof. Turski: Świadomość potrzeby szkolenia informatyków mieliśmy dużo wcześniej. Już w 1963 roku powstaje na Uniwersytecie Warszawskim sekcja, którą moglibyśmy nazwać sekcją informatyki, a wówczas nazywała się kierunkiem metod numerycznych. Zbiegło się to z zakupem komputera GIER dla powołanego wówczas do życia Zakładu Obliczeń Numerycznych. Prologomena do transakcji stanowią zresztą swojego rodzaju ciekawostkę. Otóż pierwotnie ZON przymierzał się do zakupu komputera ELLIOT.

Ale będąc w Monachium na kongresie IFIP dowiedziałem się od kogoś, kto pamiętał mnie z cyklu wykładów, który dałem swego czasu w Kopenhadze, że mógłby mi udostępnić obejrzenie komputera GIER, którego nieseryjną produkcję Duńczycy właśnie podjęli. Było to coś więcej, niż rzut oka na maszynę — nocą sprawdzałem jej działanie i to spotkanie zadecydowało o zakupie GIERA. GIER sprawdził się w praktyce zakładu. Mija 11 lat od chwili, gdy własnymi siłami zainstalowaliśmy maszynę, a ona służy nam dobrze, wychowało się przy niej wielu informatyków. Co nie znaczy, że nam wystarcza... Mimo dokonywanych w międzyczasie uzupełnień (np. urządzeniu firmy CONTROL DATA) jest to już okaz archiwalny. Trzeba zresztą podkreślić, że GIER dawno się zamortyzował.

Prof. dr hab. St. Turski kierownik Studium Informatyki UW, Kierownik Zakładu Obliczeń Numerycznych UW — pionier zastosowań informatyki w matematyce, twórca koncepcji szkolenia informatyków w trybie uniwersyteckim.

INFORMATYKA: wracając do sprawy Studium...

Prof. Turski: Mając maszynę można było sprawę szkolenia kadr dla informatyki pchnąć do przodu. Początkowo szkoliliśmy — na zasadzie specjalizacji — studentów matematyki lub pokrewnych wydziałów. Mieliśmy jednak ciągle na uwadze powołanie oddzielnego kierunku informatyki. Dyskusję wokół takiej możliwości rozpoczęły się już w roku 1969. Szansa pojawiła się w roku 1971. Wówczas to, przy okazji reformy programu szkolenia matematyków na wyższych uczelniach, inicjatywa Instytutu Maszyn Matematycznych UW została urzeczywistniona.

Należy tu podkreślić, że chętnych na kierunek informatyczny bywa zazwyczaj znacznie więcej niż miejsc. Trzeba robić selekcję. Najpopularniejsza jest specjalizacja w zakresie języków i systemów programowania.

Formalną autonomię uzyskał kierunek informatyczny z początkiem roku akademickiego 1971/1972. Od tej pory rokrocznie 75—100 studentów z grona przyjętych na wydział matematyki składa wstępną deklarację na specjalizację informatyczną. Po pierwszym semestrze drogi matematyków zaczynają się z wolna rozchodzić, po drugim — proces ten się utrwala. Z początkiem trzeciego roku monolityczny dotąd kierunek rozpada się na trzy specjalizacje: analizę numeryczną, języki i systemy programowania oraz teorię maszyn matematycznych.

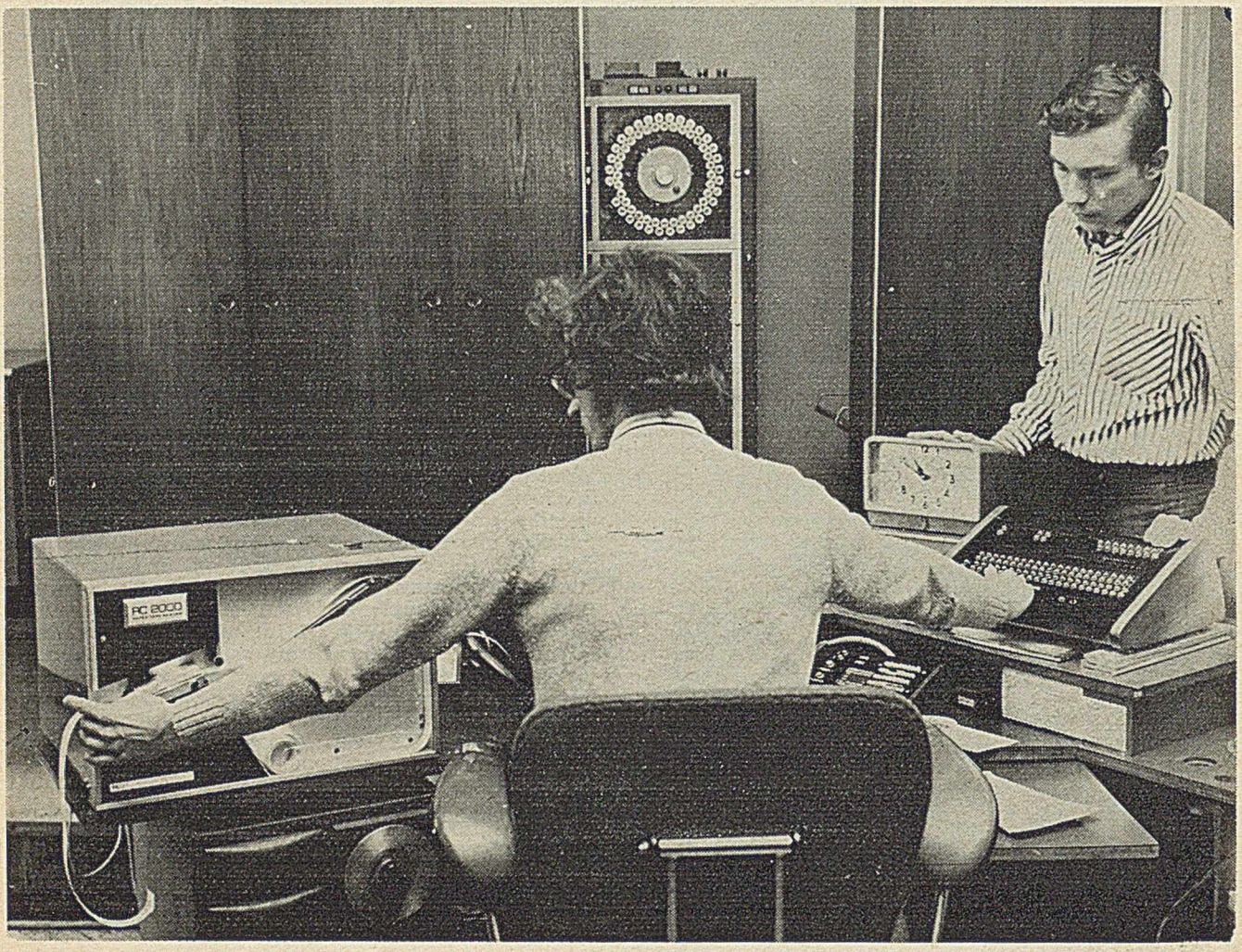
Tabela I

Program studiów na Sekcji Informatyki Wydziału Matematyki i Mechaniki Uniwersytetu Warszawskiego

Rok 1 i 2	Semestry			
	I	II	III	IV
Podstawy nauk politycznych		2—	2—	
Marksistowska filozofia i socjologia				2 2
Lektorat nauk politycznych				1—
Praktyczna nauka języka rosyjskiego	2	—2	—2	
Praktyczna nauka języków zachodnio-europejskich	—2	—2	—2	—2
Wstęp do matematyki	2e2			
Wstęp do informatyki	2 1			
Geometria z algebrą liniową	3 3	4e4		
Analiza matematyczna	4e4	4e4	4 4	4e4
Fizykalne podstawy maszyn cyfrowych		2 2		
Programowanie maszyn cyfrowych		1—		
Pracownia programowania		—3		
Algebra liniowa			2 2	
Równania różniczkowe			2e2	
Funkcje analityczne				2e2
Metody numeryczne			1	2 3
Struktury maszyn cyfrowych			2	
Pracownia programowania i struktur maszyn			—3	—3

(liczby oznaczają liczbę godzin wykładu i ćwiczeń tygodniowo, e — egzamin w końcu semestru).

Od 11 lat przyciasne pomieszczenie Zakładu Obliczeń Numerycznych wypełnia komputer GIER. Za widocznymi w głębi drewnianymi drzwiami kryje się Jednostka Centralna GIERA. Oprzyrządowanie maszyny stanowi istny konglomerat wyrobów różnych firm. Po lewej stronie czytnik taśmy papierowej RC 2000, po prawej — pulpit operatorski.



INFORMATYKA: Jakie zasady przyjęto za podstawowe?

Prof. Turski: Przyświecała nam zasada dostarczania kadry informatyków o wysokiej znajomości przedmiotu. Tego efektu nie uzyska się metodą kursów. Kursy bowiem nie dają rzeczy zupełnie podstawowej: nie dają dostatecznie głębokiej wiedzy o możliwościach maszyny.

A dobry informatyk to taki, który w czasie nauki bezustannie konfrontuje teorię z praktyką. Dobry projekt systemu — to taki, który powstaje w oparciu o dokładną znajomość możliwości maszyny. Można wtedy mówić, że wiedza empiryczna nabyta w praktycznym kontakcie z maszyną podaje koncepcje pod optymalny kształt systemu.

Dobra teoria i duża praktyka — to dwa filary wspierające zasady szkolenia informatyków. Naszą troską było i jest jak najwcześniejsze i najczęstsze udostępnienie studentom kontaktu z maszyną.

Oczywiście bariera braku sprzętu jest trudna do przełamania.

Tymczasem stały kontakt z maszyną jest konieczny także dla przyswojenia sobie wiedzy o ekonomice, o właściwym gospodarowaniu czasem maszyny. Słaby, niedouczony projektant potrzebuje znacznie więcej godzin na uruchomienie systemu, a czas maszyny jest drogi. Dlatego czasami myślę, że trudności ze sprzętem przyniosły tę jedną uboczną korzyść, że studenci nie mogli sobie pozwolić na luksus elementarnych błędów. Na maszynę „wchodziło się" z przemyślaną, dopracowaną koncepcją. Nie można wszelako tego traktować jako argumentu wobec naszego zapotrzebowania na sprzęt.

Chciałbym tu przypomnieć, że bez pomocy Centrum Obliczeniowego PAN i bez wynajmowania czasu w różnych ośrodkach — np. ZOWAR — oraz abonenckiej końcówki w systemie CYFRONET, trudności ze szkoleniem studentów byłyby prawie nie do przezwyciężenia.

Tabela II

Rok 3,4, i 5. Analiza numeryczna

	Semestry					
	V	VI	VII	VIII	IX	X
Struktury systemów cyfrowych	2e3					
Rachunek prawdopodobieństwa i statystyka	2 2	2e2				
Analiza numeryczna III	2 2	2e2				
Równania różniczkowe	2 2	2e2				
Funkcje analityczne	2e2					
Funkcje specjalne		3—	2e-			
Analiza numeryczna I	2e2	2e4				
Seminarium	—2	—2	—2	—2	—2	—2
Analiza numeryczna II			2e4	2e4		
Metody optymalizacji			2 2	2e2		
Mechanika teoretyczna			2 2	2e2		
Wykład monograficzny I			2 2	2e2		
Języki i techniki programowania					2 2	2e2
Wykład monograficzny II					2 2	2e2
Wykład monograficzny III					2 2	2e2

Tabela III

Rok 3, 4 i 5. Języki i systemy programowania

	Semestry					
	V	VI	VII	VIII	IX	X
Struktury systemów cyfrowych	2e3					
Rachunek prawd. i statystyka	2 2	2e2				
Teoria automatów i algorytmów	2 2	2e2				
Programowanie i języki progr.	2 3	2e3				
Oprogramowanie podstawowe	2 2	2e2				
Seminarium	—2	—2	—2	—2	—2	—2
Przetwarzanie danych			2 3	2e3		
Metody optymalizacji			2 2	2e2		
Pracownia			222	2e2		
Wykład monograficzny I			222	2e2		
Wykład monograficzny I	222	2e2				
Wykład monograficzny II			2 2	2e2		
Zastosowanie informatyki					2 2	2e2
Wykład monograficzny III					2 2	2e2
Wykład monograficzny IV					2 2	2e2

Tabela IV

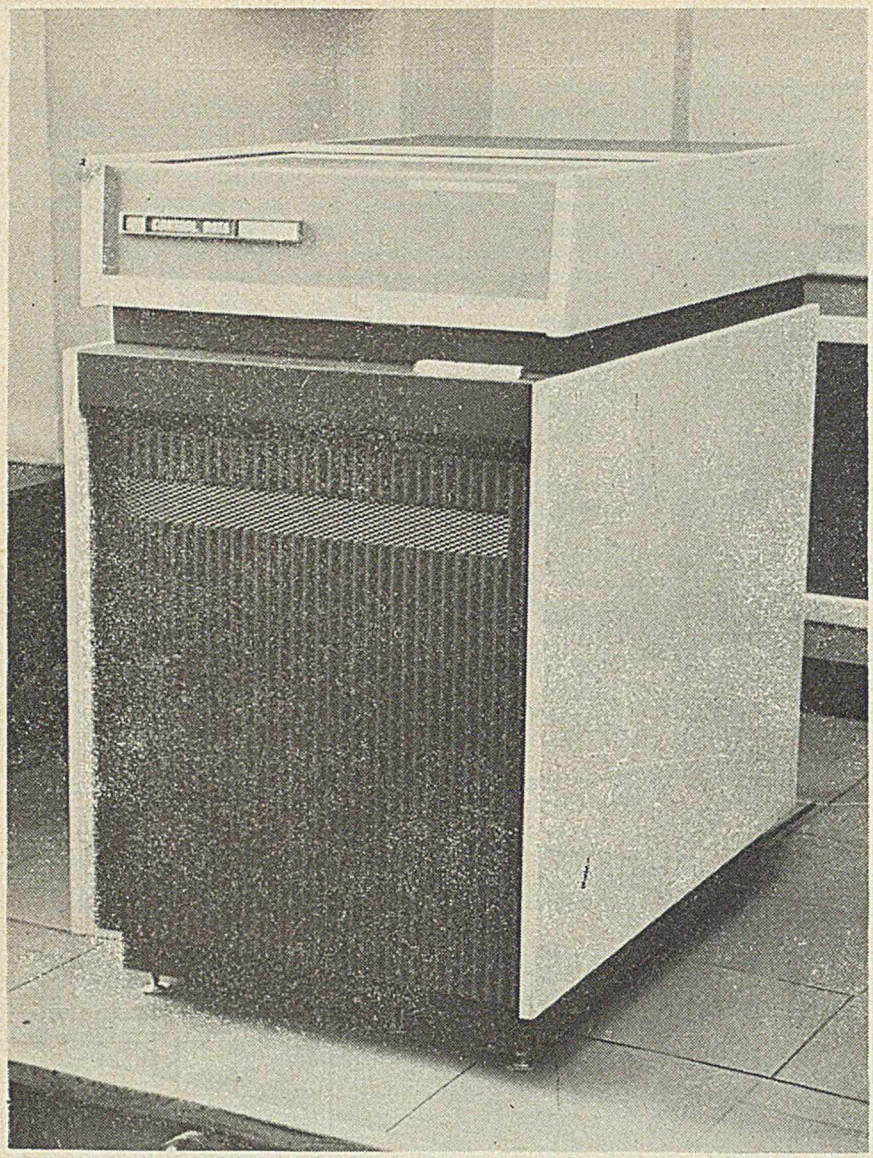
Rok 3, 4 i 5. Teoria maszyn matematycznych.

	Semestry					
	V	VI	VII	VIII	IX	X
Struktury systemów cyfrowych	2e3					
Rachunek prawdopodobieństwa i statystyka	2 2	2e2				
Teoria automatów i algorytmów	2 2	2e2				
Programowanie i języki programowania	2 2	2 2				
Logika i logika algorytmiczna	2 2	2e2				
Seminarium	—2	—2	—2	—2	—2	—2
Metody teorii grafów		2e2				
Teoria systemów			2 2	2 2		
Zastosowanie informatyki			—3	—3		
Pracownia			—3	—3		
Wykład monograficzny I			2 2	2e2		
Wykład monograficzny II			2 2	2e2		
Wykład monograficzny III			—	—	2 2	2e2
Wykład monograficzny IV					2 2	2e2

INFORMATYKA: W jakim zakresie uczelnia zamierza wyposażyć absolwentów Studium w wiedzę o informatyce?

Prof. Turski: Program nauczania dla studentów informatyki powstawał w oparciu o analizę programów stosowanych na uczelniach zagranicznych. Zakłada on nawiązanie kontaktu z maszyną już na II semestrze (a więc praktycznie na samym początku szkolenia przyszłego informatyka) i nauczenie się w tym samym semestrze programowania w języku ALGOL; na II roku studiów — w języku LISP, w III — w języku FORTRAN, w IV — w językach COBOL i SIMULA. Solidne podstawy dają wykłady: z zakresu fizykalnych podstaw maszyn cyfrowych, struktury maszyn cyfrowych, analizy numerycznej i metod numerycznych, teorii maszyn matematycznych. Sądzimy, że obraliśmy właściwą koncepcję nauczania. Mamy już zresztą spore doświadczenie wyniesione z okresu przed powstaniem kierunku. Wówczas również szkoliliśmy informatyków, nie tak wielu i w nie tak szerokim zakresie, ale dość dobrze. Absolwenci UW legitymujący się specjalizacją informatyczną jeszcze nie objętą mianem Studium byli dobrze widziani — nie tylko w kraju. Wielu zresztą później doktoryzowało się u nas.

Oczywiście — zdajemy sobie sprawę, że można uczyć jeszcze lepiej. Ale informatyki uczyć należy — powtarzam to z uporem — na dobrym sprzęcie. Ma go przemysł, ośrodki branżowe, a my — mamy dobrą kadrę dydaktyczną, chętnych studentów, sensowny program nauczania i prócz — znużonego już GIERA — żadnego sprzętu. Rozbudowa Zakładu Obliczeń Numerycznych pozwoliłaby szkolić większą rzeszę tak poszukiwanych dobrych informatyków. Należy przy tym pamiętać, że ZON — jednostka pozostająca poza strukturą Wydziału Matematyki, podlegała bezpośrednio Rektorowi (chociaż kierowana przeze mnie), ma do spełnienia wiele różnych funkcji. Właściwie Za-



Nowoczesna jednostka dyskowa produkcji CONTROL DATA. Tylko systematyczne uzupełnienie zestawu GIER pozwoliły mu sprostać zwiększającym się stale zadaniom.

kład powinien być do dyspozycji całej uczelni. Tymczasem — z konieczności punkt ciężkości całej działalności ZON-u przesuwa się w stronę dydaktyki. Wzmocnienie mocy obliczeniowej ZON-u byłoby korzystne z innego także względu. Mówiło się już o kadrze naukowej Studium ZON-u, o możliwościach studentów. ZON oddał już spore usługi gospodarcze, mógłby je zwiększyć. Szeroki kontakt studentów z maszyną umożliwiłby im podejmowanie konkretnych zadań na rzecz różnych insytucji. A przy tym praktyka wykazała niedostateczność masowych form szkolenia informatyków stosowanych w Polsce.

Nie wzbudza chyba wątpliwości uwaga, że zainstalowane w kraju maszyny nie są optymalnie wykorzystane z powodu niskiego stopnia przygotowania kadry. Natomiast metoda 4,5 lub 5-letniego specjalistycznego szkolenia od podstaw, przyjęta przez Uniwersytet Warszawski, nie wzbudza zastrzeżeń. Należało by ją upowszechnić (śladem UW podążyły uczelnie Dolnego Śląska) zwłaszcza, że informatyka jest wśród młodzieży licealnej dyscypliną popularną, o czym świadczy zresztą duża ilość zgłoszeń na ten kierunek.

INFORMATYKA: Jakie są perspektywy dalszego rozwoju Studium Informatyki?

Prof. Turski: Myślę, że ze wszech miar pożytecznym było by przekształcenie Studium z kierunku na samodzielny Wydział Informatyki. Decyzja pozornie tylko administracyjna miała by szerokie implikacje, przede wszystkim w zakresie sprzętu i środków na eksploatację. Z kolei odpowiednia baza sprzętowa pozwoliłaby przyjmować na studia informatyczne znacznie większą liczbę studentów.

Oczywiście, pociągnęłoby to za sobą konieczność zwiększenia naszej kadry dydaktycznej. Ta, którą dysponujemy, zasługuje na wysokie uznanie. Ale byłoby nam ciążko, gdyby nie współpraca takich naukowców, jak prof. Z. Pawlak, lub prof. L. Łukaszewicz. Mamy też sporą grupę bardzo dobrych asystentów. Sądzę, że u kresu mojej pracy dydaktycznej uda mi się spełnić zamierzenie zbudowania solidnych podstaw do szkolenia kadry informatyków. Chcę tu stwierdzić, że pomimo licznych, cytowanych kłopotów, klimat dla rozwoju nauczania informatyki na UW jest więcej, niż dobry. Kłopoty zaś z upływem czasu postaramy się przezwyciężyć.

Rozmawiał: **Krystyn Bernatowicz**

DOROTA PRAWDZIC
INFORMATYKA

681.322.004.14:65.011.56(47)

ZSRR: postępy automatyzacji zarządzania

Przedstawiono program perspektywiczny rozwoju informatyki w gospodarce radzieckiej i omówiono realizację planów wdrażania zautomatyzowanych systemów zarządzania w bieżącym pięcioleciu.

W Związku Radzieckim lat 1966—70 uruchomiono 417 systemów komputerowych, w tym blisko 300 zautomatyzowanych systemów zarządzania. Zorganizowano w tym czasie ponad 700 ośrodków obliczeniowych. Był to jeszcze okres wysiłków mało koordynowanych i raczej eksperymentalnych, borykano się z przeróżnymi trudnościami, przełamywano bariery techniczne, organizacyjne, psychologiczne. Tym lepiej rysują się obecne pozytywne strony wielkiego rozruchu informatyki radzieckiej. Przede wszystkim powstał i rozwinął się przemysł sprzętu informatyki, a na pierwszy plan wśród komputerów tego okresu wysunęły się — dziś już zasłużone — MIŃSK-22, następnie MIŃSK-32 — maszyny II generacji, ze stale rozwijanym oprogramowaniem umożliwiającym tworzenie zautomatyzowanych systemów zarządzania.

Doświadczenia z wprowadzania wielu różnorodnych systemów informatycznych w przedsiębiorstwach i branżach wykazały, że jest ono najbardziej celowe tam, gdzie organizacja produkcji stoi na wysokim poziomie, gdzie są starannie dopracowane procesy technologiczne, gdzie zwraca się uwagę na uporządkowanie normatywów.

Pionierskie zautomatyzowane zakładowe systemy zarządzania wprowadzono m.in. w Lwowskiej Fabryce Telewizorów, w Mińskiej Fabryce Ciągników, w Barnaulskich Zakładach Radiotechnicznych. Powstały też pierwsze branżowe zautomatyzowane systemy zarządzania, jak ASU-Pribor w Ministerstwie Przemysłu Przyrządów Pomiarowych, Automatyki i Systemów Sterowania.

Pierwsi projektanci systemów informatycznych wzięli na siebie cały ciężar torowania dróg, zwalczania trudności, a przede wszystkim zbierania wszystkich doświadczeń.

W skład grup projektantów wchodzili przedstawiciele takich placówek naukowo-badawczych, jak Instytut Cybernetyki Akademii Nauk USRR, Centrum Obliczeniowe Syberyjskiej Filii Akademii Nauk ZSRR, Centralny Naukowo-Badawczy i Projektowo-Technologiczny Instytut Zarządzania i wiele innych. W projektowaniu i wdrażaniu systemów aktywnie uczestniczyli przedstawiciele użytkowników — zakładów i resortów.

Foto 1. Taszkient — Fabryka Kabli. Pulpit sterowniczy urządzeń walcowni miedzi

I chociaż wachlarz rozwiązywanych problemów był dość ograniczony, a wyposażenie techniczne często niedostateczne — systemy informatyczne okazały się opłacalne; zwrot nakładów następował już po 2—3 latach.

Zebrane doświadczenia i wnioski umożliwiły władzom partyjnym i państwowym sformułowanie programów dalszego rozwoju informatyki w ZSRR, a rozwój informatyki ściśle związano z usprawnianiem zarządzania gospodarką narodową.

W zakresie automatyzacji zarządzania planuje się stworzenie ogólnopaństwowego zautomatyzowanego systemu zbierania i przetwarzania informacji do ewidencji, planowania i zarządzania gospodarką narodową ZSRR. System ma się opierać na państwowej sieci ośrodków obliczeniowych i na jednolitej sieci łączności.

System ten (OGAS) ma objąć zautomatyzowane systemy zarządzania przedsiębiorstwami (ASUP) oraz zjednoczeniami wielozakładowymi (ASUO), branżowe zautomatyzowane systemy zarządzania (OASU), jak również systemy o znaczeniu ogólnym, takie jak: zautomatyzowany system statystyki państwowej (ASGS), system planowania (ASPR), system informacji i zarządzania normami (AISU), system przetwarzania informacji o cenach (ASOJ-Cen), system sterowania postępem naukowo-technicznym (ASUNT), systemy bankowe itd. Projektuje się też systemy informatyczne dla republik związkowych, większych miast, jednostek administracyjnych różnych szczebli.

Inż. DOROTA PRAWDZIC ukończyła Wydział Elektryczny Politechniki Warszawskiej (1949). Pracownik przemysłu elektronicznego i półprzewodnikowego do 1960 r. Kierownik Branżowego Ośrodka Informacji INTE Instytutu Maszyn Matematycznych w Warszawie (1961—1973). Inicjator i organizator licznych wydawnictw książkowych, seryjnych i periodycznych z dziedziny informatyki oraz autor wielu artykułów i opracowań popularno-naukowych. Współzałożyciel czasopism ETO-NOWOŚCI i INFORMATYKA. Zastępca redaktorów naczelnych tych periodyków.

Duże znaczenie przywiązuje się do wprowadzania systemów automatycznego sterowania procesami technologicznymi (ASUTP). Założone tempo wzrostu przewiduje, że w latach 1971—75 zbuduje się 6—7 razy więcej systemów ASUTP, niż w poprzednim pięcioleciu 1966—70.

Do końca bieżącego pięciolecia wprowadzi się ponad 2000 zautomatyzowanych systemów zarządzania (ASU) oraz 700 systemów automatycznego sterowania procesami technologicznymi (ASUTP). Inwestycje na rozwój informatyki w ZSRR przekroczą 6 mld rubli. Na prace naukowe i projektowe przeznacza się 1,3 mld rubli. Potrzeba do tego ok. 150 tys. specjalistów. W wyniku licznych konferencji i narad poświęconych zautomatyzowanym systemom zarządzania w przedsiębiorstwach i branżach przemysłowych, szczególny nacisk położono na koordynację działań oraz na typizację projektów.

Komitet Nauki i Techniki Rady Ministrów ZSRR opracował skorygowany pięcioletni plan rozwoju sprzętu i zastosowań informatyki. Sporządzono również jednolity plan finansowania rozwoju prac naukowo-badawczych w tej dziedzinie. Wspólnie z Ministerstwem Przemysłu Przyrządów Pomiarowych, Automatyki i Systemów Sterowania, Ministerstwem Przemysłu Radioelektroniki i Państwowym Komitetem Planowania — Komitet Nauki i Techniki ZSRR opracował i wprowadził jednolitą formę założeń projektowych dotyczących tworzenia branżowych zautomatyzowanych systemów zarządzania (OASU), a następnie zatwierdził typowy projekt techniczny OASU. Opracowano również typowy projekt wyposażenia technicznego branżowych ośrodków obliczeniowych. Przygotowano materiały dotyczące projektów roboczych niektórych branżowych zautomatyzowanych podsystemów zarządzania, między innymi podsystemów perspektywicznego rozwoju branży, planowania techniczno-ekonomicznego, ewidencji księgowej, zaopatrzenia materiałowo-technicznego, zbytu.

Opracowano szereg innych wytycznych, takich jak: „Ogólnobranżowe wytyczne metodologiczne do tworzenia zautomatyzowanych systemów zarządzania przedsiębiorstwami (ASUP)".

Dużą uwagę zwrócono na metody obliczenia ekonomicznej efektywności zautomatyzowanych systemów zarządzania przedsiębiorstwami (ASUP) i branżami (OASU). Opracowane metody potraktowano jako tymczasowe, zdając sobie sprawę z trudności problemu i konieczności praktycznego sprawdzenia metodologii.

Prace nad realizacją zautomatyzowanych systemów zarządzania (ASU) biegną obecnie bardzo szerokim frontem. Liczba wdrożonych systemów informatycznych w dniu 1 stycznia 1974 r. przekroczyła 1200. W bieżącym pięcioleciu ma powstać łącznie 2700 systemów, w tym systemów zarządzania branżowych — 200, zakładowych — 1800, systemów sterowania procesami technologicznymi — 700. Zorganizowanych zostanie 2200 ośrodków obliczeniowych.

Baza techniczna ASU

Podstawowym komputerem, szeroko stosowanym w bieżącym pięcioleciu w zautomatyzowanych systemach zarządzania, ma być MIŃSK-32. Liczbę zainstalowanych obecnie w ZSRR komputerów tego typu szacuje się na około 3 tysiące.

W związku z uruchamianiem produkcji komputerów i urządzeń zewnętrznych Jednolitego Systemu RIAD [1], postanowiono powiązać ze sobą sprzęt bieżąco produkowany z nowym przenosząc dorobek programowy komputerów MIŃSK-32 na maszyny Jednolitego Systemu. Ponadto opracowuje się nowe urządzenia wymiany informacji między komputerem MIŃSK-32, a urządzeniami zewnętrznymi co umożliwi przyłączanie wszystkich urządzeń zewnętrznych JS EMC do komputerów typu MIŃSK. Produkcja urządzeń zewnętrznych jest zresztą traktowana jako równie ważna, jak produkcja samych komputerów. Jednolity System RIAD zawiera — obok komputerów różnej wielkości — około 150 urządzeń zewnętrznych. Radzieccy użytkownicy komputerów wysuwają postulaty dostarczania do ośrodków obliczeniowych kompletnych zestawów urządzeń, jak również zaopatrzenia w dobry nowoczesny sprzęt do rejestracji danych. Żądają też polepszenia jakości taśmy magnetycznej, papierowych nośników informacji i innych materiałów eksploatacyjnych.

Foto 2. Psków — Instytut Rozwojowy Techniki Obrabiarek. MIŃSK 32

Zautomatyzowane systemy zarządzania ściśle łączą się z automatycznymi systemami sterowania procesami technologicznymi, szczególnie w takich gałęziach przemysłu, jak hutnictwo, energetyka, górnictwo, chemia, petrochemia, gazownictwo.

Przygotowywaniem systemowego zastosowania automatyki w ZSRR zajmuje się resort przemysłu przyrządów pomiarowych, automatyki i systemów sterowania. Od 1973 r. zakłady podległe Ministerstwu Przyrządów Pomiarowych, Automatyki i Systemów Sterowania rozpoczęły dostawę nowych modułowych cyfrowych systemów III generacji ASWT M6000 i M4000. Doświadczenia z ich wdrażania wykazały, że istnieją obiektywne trudności z przekwalifikowaniem odbiorców systemów i projektantów, którzy wychowali się na komputerach II generacji o stałej, niemodularnej strukturze (MIŃSK 22, MIŃSK 32, DNIEPR 1 i 2, UM-1 itd.). Ocenia się, że trudności powinny być przezwyciężone w ciągu najbliższych 1—2 lat. Obecnie trwają prace nad dalszym rozwojem tego t.zw. kompleksu sterująco-obliczeniowego (UWK) — nad wzbogacaniem wachlarza modeli, przechodzeniem na zestawy wieloprocesorowe, doskonaleniem struktur i parametrów techniczno-eksploatacyjnych oraz nad typizacją i unifikacją oprogramowania.

[1] Na produkcję tę nastawia się wiele zakładów. Jest już uruchomiona produkcja seryjna komputerów EC-1020 (R-20) i maszyny te są dostarczane na rynek wewnętrzny i na eksport, m.in. zainstalowano w 1974 r. w Polsce kilka tych maszyn.

Etapy wprowadzania ASU

Wytyczne Komitetu Nauki i Techniki Rady Ministrów ZSRR zaleciły przedsiębiorstwom etapowe wdrażanie ASUP. W I etapie powinno się tworzyć takie podsystemy, jak:

— techniczne przygotowanie produkcji
— planowanie techniczno-ekonomiczne
— planowanie produkcji i ewidencja podstawowych wyrobów
— zaopatrzenie techniczno-materiałowe
— zbyt wyrobów gotowych
— ewidencja księgowa
— ewidencja kadr.

W zasadzie są to podsystemy ewidencyjne (planistyczno-sprawozdawcze), poza może podsystemem technicznego przygotowania produkcji, który często obejmuje automatyzację prac projektowych i obliczenia inżynierskie związane z konstrukcją wyrobów i technologią produkcji. Stwierdzono, że są one bardzo przydatne przy opracowywaniu planów długoterminowych i bieżących na szczeblu zakładów i instytucji nadrzędnych. Ale nie są to oczywiście pełne korzyści komputeryzacji. Dopiero na drugim jej etapie spodziewana jest możliwość głębokiego analizowania informacji i poszukiwania optymalnych decyzji dotyczących kompleksowej działalności jednostek gospodarczych i branż. Toteż zaleca się, aby kolejne i stopniowe wprowadzanie do eksploatacji poszczególnych ogniw ASU było poprzedzane opracowaniem ogólnego całkowitego projektu systemu. Podsystemy ASU powinny odpowiadać jednolitej zasadzie organizacji całego systemu, a przede wszystkim powinny opierać się na jednolitej wspólnej bazie informacyjnej. Te przedsiębiorstwa, w których projektowano systemy komputerowe jako luźne zbiory zautomatyzowanych czynności w zakresie zarządzania i to na ogół arytmetycznych, nie potrafiły efektywnie wykorzystać komputerów i są dalekie od możliwości wprowadzenia ASU w najbliższym czasie.

Coraz większe znaczenie przywiązuje się do technologii zautomatyzowanego przetwarzania danych i jej normalizacji. Prace Komitetu Nauki i Techniki ZSRR idą również i w tym kierunku. Problem organizacji zbierania i przekazywania na odległość informacji technicznych i ekonomicznych traktowany jest obecnie, jako równorzędny z organizacją samego przetwarzania. W Związku Radzieckim niewielkie są jeszcze doświadczenia ze stosowaniem teletransmisji danych. Toteż specjalną opieką otacza się systemy teleinformatyczne rozwijane w takich instytutach, jak Państwowy Komitet Planowania, Centralny Urząd Statystyczny, Bank Państwowy.

W celu lepszego skoordynowania prac nad projektowaniem i uruchamianiem ASU, którymi zajmuje się obecnie w ZSRR ok. 50 tysięcy osób, powołano Międzyresortową Radę do Usprawniania Zarządzania Gospodarką Narodową. Jako jedno z ważniejszych — Rada podjęła zadanie poszukiwania typowych rozwiązań dla rozmaitych aspektów organizacji ASU, w tym również budowy banków informacji. Oczywiście nie ma się przy tym na celu likwidacji różnorodności funkcjonalnych czynności administracyjnych, jednakże dąży się do ustalania wspólnych dla wszystkich funkcji zarządzania sposobów i metod przetwarzania informacji, jak również sposobów wykorzystywania informacji wynikowych i zbiorczych. Ponieważ i tak część prac będzie musiał wykonywać człowiek, więc poszukuje się prawidłowych proporcji między operacjami zautomatyzowanymi i niezautomatyzowanymi.

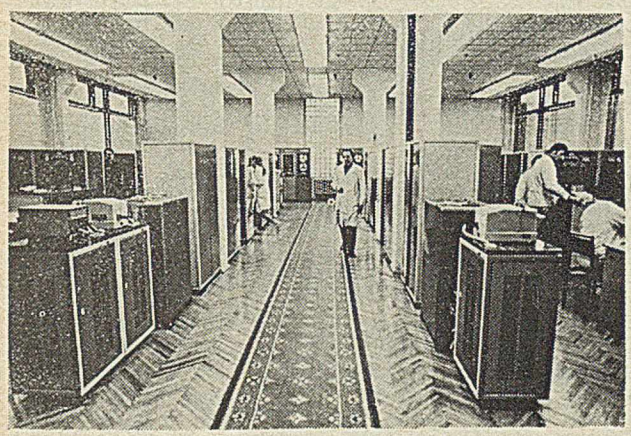


Foto 3. Moskwa — Fabryka Łożysk. Sala komputerów w ośrodku obliczeniowym

Foto 4. Azerbajdżan — Centrum Obliczeniowe Urzędu Statystycznego. Sala przygotowywania danych

Przykłady ASU funkcjonujących w zakładach produkcyjnych

Do zakładów przemysłowych wyróżniających się w dziedzinie funkcjonowania zakładowych systemów (ASUP) należą: Zakłady Produkcji Narzędzi FREZER w Moskwie, II Fabryka Zegarków w Moskwie, Leningradzkie Zjednoczenie Przemysłu Optyczno-Precyzyjnego (przedsiębiorstwo wielozakładowe), Wileńskie Zjednoczenie SIGMA Przemysłu Maszyn Liczących (przedsiębiorstwo wielozakładowe), Fabryka Ciągników i Fabryka Zegarków w Mińsku, Donieckie Zakłady Przemysłu Maszynowego im. Leninskiego Komsomołu Ukrainy, Lwowska Fabryka Telewizorów, URALMASZZAWOD, Barnaulskie Zakłady Radiotechniczne i wiele innych.

Na przykład prace nad ASUP w Mińskiej Fabryce Ciągników trwają od 1965 r. I etap ASUP wprowadzono do eksploatacji w 1969 r. Zorganizowano ośrodek obliczeniowy wyposażony w 3 komputery MIŃSK-22. Uporządkowano dokumentację konstrukcyjną, technologiczną, planistyczną i sprawozdawczą. Opracowano system kodowania informacji, stworzono gospodarkę normatywną w postaci kartotek zapisanych na taśmach magnetycznych (obejmuje ona wykazy części konstrukcyjnych, dane przebiegów technologicznych, normy zużycia materiałów, pracochłonności i inne normatywy).

Zapewniono aktualizację normatywów i klasyfikacji oraz wiarygodność danych. Przeszkolono kadrę programistów, elektroników, operatorów, ekonomistów. Opracowano ponad 350 programów komputerowych. Wdrożono 5 podsystemów (techniczne przygotowanie produkcji, planowanie techniczno-ekonomiczne, operatywne planowanie podstawowej produkcji, zaopatrzenie materiałowo-techniczne i ewidencję kadr).

System funkcjonuje w dwóch płaszczyznach: międzywydziałowo i wewnątrzwydziałowo (w 6 wydziałach produkcyjnych). Przy poniesionych dotąd nakładach

Foto 5. Lwów — Fabryka Telewizorów. Hala produkcyjna

w wysokości 2 mln rb uzyskano efekty ekonomiczne wartości ponad 1 mln rb. rocznie, dzięki wzrostowi wydajności, zmniejszeniu przestojów maszyn, wzrostowi rytmiczności produkcji, wykryciu wielu rezerw i wskutek nowego podejścia do wielu funkcji zarządzania.

Od 1971 r. opracowuje się II etap ASUP — kompleksowy system planowania, ewidencji i regulowania produkcji podstawowej we wszystkich wydziałach produkcyjnych.

O podniesieniu dyscypliny produkcji, ułatwieniach pracy i zwiększeniu się roli mistrzów w organizacji produkcji, w wyniku ponad dwuletniego działania zautomatyzowanego systemu zarządzania w Barnaulskich Zakładach Radiotechnicznych, pisze korespondent Ekonomiczeskoj Gaziety (1973 r., nr 44).

Nastąpiło tam uporządkowanie obiegu dokumentów, podstawowym dokumentem produkcyjnym stała się karta normatywno-technologiczna. Mistrzowie nie zajmują się już zbędną pisaniną, zarówno oni, jak i robotnicy otrzymują zadania wyliczone w ośrodku obliczeniowym. Sprawozdania dzienne o wykonaniu zadań są na bieżąco udostępniane w miejscach pracy, wraz z informacjami o zużyciu materiałów, o zaawansowaniu obróbki, a nawet o naliczeniu zarobków. Ocenia się, że dzięki ASUP nie tylko kierownictwo wydziałów, ale i robotnicy wciągają się w proces planowania produkcji.

Na rozwój Zakładów Produkcji Maszyn Matematycznych im. Ordżonikidze w Mińsku w dużej mierze — wedle oceny dyrekcji tych Zakładów — wpłynęło wdrożenie I etapu ASUP. W ciągu lat 1971—72 produkcja wzrosła dwukrotnie przy 1,5-krotnym wzroście wydajności.

Dyrekcja Mińskiej Fabryki Zegarków stwierdza, że dzięki wdrożeniu I etapu ASUP uzyskano znaczne zmniejszenie produkcji w toku, zwiększyła się rytmiczność pracy, zmniejszono zaangażowanie środków obrotowych o około 1 mln rb. Roczne efekty ekonomiczne szacuje się na ok. 0,5 mln rb. Zmniejszyła się nawet liczba zatrudnionych pracowników administracyjnych. Podobne efekty podaje kierownictwo II Moskiewskiej Fabryki Zegarków.

Przykłady ASU w skali republik związkowych

Prace nad wdrażaniem ASU są planowane i nadzorowane również w przekroju republik.

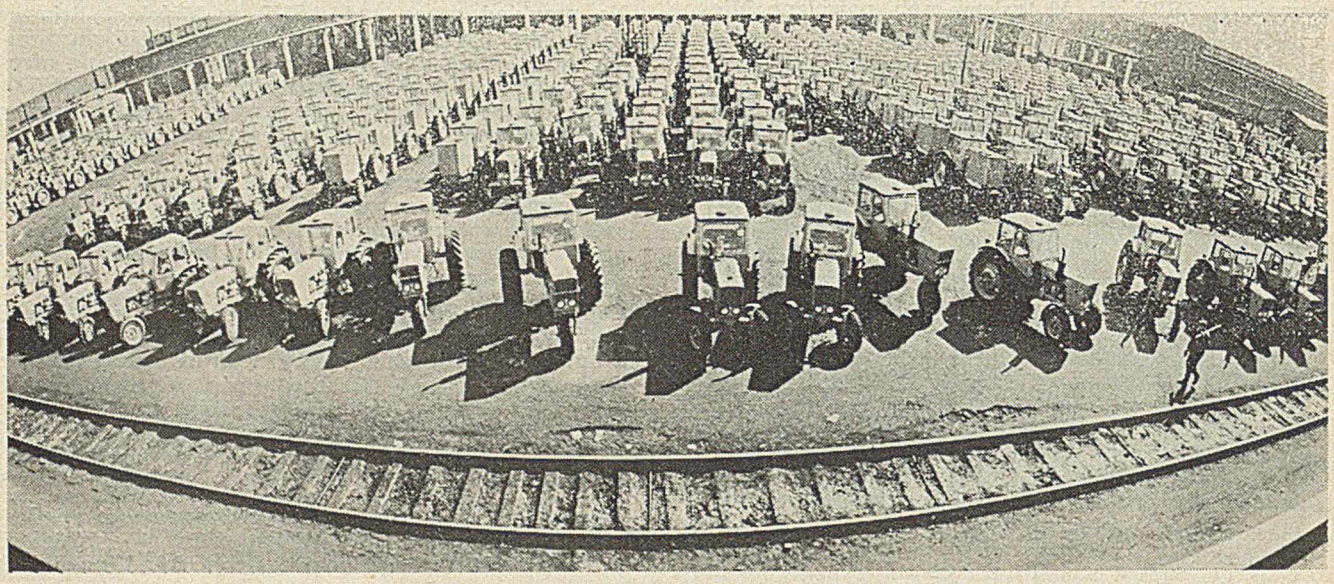
Na przykład, w Uzbekistanie przewiduje się zorganizowanie w latach 1971—75 29 branżowych, 11 przemysłowych, 10 terenowych i 5 budowlanych ASU, a oprócz tego — 58 ośrodków obliczeniowych w zakładach produkcyjnych, placówkach naukowo-badawczych i projektowych. Nakłady na te inwestycje, łącznie z pracami projektowymi, będą wynosić ok. 100 mln rb. Obecnie działają tam ośrodki obliczeniowe w Gosplanie, Urzędzie Statystycznym oraz w 9 ministerstwach i urzędach Uzbeckiej SRR.

W Białoruskiej SRR pracuje 48 ośrodków obliczeniowych, wdrożono 32 różnego rodzaju zautomatyzowane systemy zarządzania.

Między innymi, w transporcie samochodowym — dzięki ASU — uzyskano poważne, ok. 10-procentowe zwiększenie wskaźnika wykorzystania parku samo-

c.d. na str. 17

Foto 6. Mińsk — Mińska Fabryka Traktorów



WŁADYSŁAW M. TURSKI
Instytut Maszyn Matematycznych Zjednoczenia MERA
Wydział Matematyki i Mechaniki UW

681.322.06.001:8

Główne kierunki badawcze w dziedzinie oprogramowania*)

Wyszczególniono kierunki badań naukowych w dziedzinie oprogramowania zgrupowane w następujących czterech tematach: systemy operacyjne, banki danych, translatory, komunikacja człowiek — maszyna.
Przedstawiono również zagadnienie uniwersalne dla tej dziedziny, mianowicie technologię wytwarzania oprogramowania i metodykę jego weryfikacji.

Oprogramowanie maszyn cyfrowych stanowi obecnie autonomiczną dzedzinę badań naukowych „o pełnym cyklu rozwojowym": od dociekań podstawowych, ściśle związanych z naukami matematycznymi (lingwistyka formalna, algebra uniwersalna, kombinatoryka analityczna, teoria obliczalności), poprzez prace konstrukcyjne, związane z naukami systemowymi (teoria systemów, optymalizacja, badania operacyjne) i prace o charakterze technologicznym (automatyzacja wytwarzania oprogramowania i jej instrumentarium, prakseologia działalności zespołowej), aż do metodologii wdrożeń i badań jakości (z badaniem niezawodności na czele).

Szerokość frontu badań w dziedzinie oprogramowania, znaczny stopień powiązań wewnętrznych pomiędzy poszczególnymi kierunkami badawczymi, wyrażający się m.in. tym, że te same problemy stanowią niezależny przedmiot studiów i, jednocześnie, pojawiają się jako zagadnienia wtórne przy innych badaniach, brak ukształtowanych, a tym bardziej dostatecznie uniwersalnych teorii, niedostateczna systematyka przedmiotów badawczych oraz całkowity rozgardiasz dydaktyczny, uniemożliwiający zadowalające kształcenie kadry naukowej dla tej dziedziny, połączone z wyraźnie wzrastającym zapotrzebowaniem społecznym na konkretne rezultaty o praktycznej użyteczności czynią badania w dziedzinie oprogramowania wysoce atrakcyjnym i niezmiernie ważnym działem informatyki, wymagającym szczególnego poparcia ze strony dysponentów środków stymulujących rozwój badań i szczególnej troski ze strony środowiska naukowego.

W niniejszym referacie Sekcja Oprogramowania przedstawia Komitetowi Informatyki wykaz najważniejszych kierunków badawczych, wybranych ze względu na dwa jednakowo ważne kryteria

- znaczenie praktyczne spodziewanych wyników badań
- względne wyodrębnienie się i ugruntowanie w całokształcie problematyki badawczej.

Kryterium pierwsze nie wymaga komentarza, kryterium drugie wybrane zostało jako rękojmia (stosunkowo) małego ryzyka decyzji koncentracji prac badawczych wokół danej tematyki.

*) Referat Sekcji Oprogramowania przygotowany na posiedzenie Komitetu Informatyki PAN, na podstawie materiałów opracowanych przez J. Borowca, W. Mardala, J. Olszewskiego, W. M. Turskiego i J. Wierzbowskiego

Sekcja Oprogramowania wnioskuje, by Komitet Informatyki PAN zechciał postulować preferowanie tych prac naukowych w dziedzinie oprogramowania, które spełniają powyższe dwa kryteria.

W ujęciu przedmiotowym kierunki badawcze grupują się w czterech tematach:

1. Systemy operacyjne
2. Banki danych
3. Translatory
4. Komunikacja człowiek—maszyna.

Omówmy pokrótce powyższe zagadnienia.

Systemy operacyjne

Dziedzina systemów operacyjnych jest najbardziej rozległą w oprogramowaniu maszyn, a w Polsce chyba najmniej uprawianą dyscypliną badawczą. Obejmuje ona oprogramowanie, które w stosunku do programów tzw. użytkowych, a także translatorów (kompilatorów, interpretatorów i generatorów) języków programowania pełni rolę zespołu programów usługowych, z drugiej zaś strony umożliwia sterowanie procesami liczenia, transmitowanie danych, komunikację człowiek—maszyna i prowadzenie wszelkich potrzebnych księgowości i statystyk. Zwróćmy uwagę, że dla niektórych zastosowań maszyn całość oprogramowania może nosić miano systemu operacyjnego. Takim zastosowaniem jest np. system rezerwacji miejsc lotniczych czy kolejowych. W zasadzie nie ma tam mowy o programach użytkowych czy translatorach języków programowania. Jest za to wiele programów (procedur, podprogramów) transmisji i konwersji danych przekazywanych pomiędzy urządzeniami końcowymi a pamięcią systemu, programów służących do organizacji, tworzenia, powiększania i aktualizacji zbiorów danych, programów redagowania wszelkich potrzebnych zestawień, sprawozdań, kosztów itp., oraz programów z góry przewidzianych na przypadek zmian w konfiguracji samego systemu (więcej i, być może, innych urządzeń końcowych, większa pamięć, większa liczba kanałów transmisji itp.).
Powyższe stwierdzenia należy traktować jako bardzo pobieżną i wstępną ilustrację następującej próby określenia głównych kierunków badawczych w dziedzinie systemów operacyjnych.

Język operowania systemem liczącym

Pod tym hasłem należy rozumieć prace w kierunku sformułowania możliwie ogólnego języka operowania systemem liczącym, tzn. języka, w którym będzie możliwe i niezbyt trudne wyrażenie tego co, kiedy i jak ma być wykonywane. Obecny stan tego kierunku najlepiej chyba ilustruje język rozkazowy *(Comandand Language)* systemów operacyjnych GEORGE dla maszyn firmy ICL, który z jednej strony uważany jest za najlepiej do tej pory sformułowany język operowania systemem liczącym, a z drugiej stro-

ny ma się tak do zamierzonego języka operowania systemem liczącym, jak kod wewnętrzny jakiejś maszyny do ALGOLu.

Z tego punktu widzenia system operacyjny powinien być traktowany jako interpretator czy nawet kompilator języka operowania systemem liczącym, co stanowi niejako odwrócenie dotychczasowej praktyki, w której język taki wynika z konstrukcji systemu operacyjnego.

Metody implementacji systemu operacyjnego

Problem ten obejmuje cały zespół zagadnień badawczych, jak:

— metody projektowania
— metody programowania (w tym próby w kierunku odpowiedniego języka programowania)
— metody wiązania z istniejącym już oprogramowaniem (w tym również z działającymi już częściami systemu operacyjnego)
— metody dokumentacji
— metody weryfikacji
— metody wprowadzania zmian wynikających z rozbudowy i zmian sprzętu oraz ze zmian potrzeb użytkowników.

Wiele, aczkolwiek nie wszystkie, z wymienionych problemów znajduje obecnie wyraz w dążeniach do sformułowania języka implementacji systemów operacyjnych w oparciu o język PASCAL, który okazał się już językiem odpowiednim dla dokumentacji niektórych części systemu operacyjnego oraz dla przekazywania wiedzy o ich budowie. W praktyce, udane próby w kierunku języka implementacji systemów operacyjnych przeprowadzono w oparciu o język ALGOL w firmie BURROUGHS, gdzie język ten odpowiednio rozszerzono, a ponadto sam sprzęt skonstruowano tak, by możliwie najpełniej odwzorować struktury danych tego języka i jego translatorów w technikę adresowania maszyny.

Metody optymalizacji działania systemu liczącego

Kierunek ten z jednej strony doczekał się już wielu opracowań zarówno praktycznych, jak i teoretycznych, obejmujących problemy i techniki podziału czasu procesorów (jednostek arytmetycznych i kanałów transmisji) i podziału pamięci. Z tego właśnie kierunku wywodzą się zaawansowane prace nad teorią i implementacją procesów współbieżnych, nad metodami ich synchronizacji oraz nad pamięciami wirtualnymi. Z drugiej strony właściwie nie umiemy jeszcze dać rzetelnej odpowiedzi na pytanie dotyczące jakości systemu operacyjnego, czy też szerzej, systemu liczącego. Jest tak dlatego, że nie potrafimy, jak dotychczas, traktować łącznie problemów optymalizacji działania systemu liczącego, widzianych z różnych punktów widzenia, tak by z tego bezpośrednio wynikały wnioski dotyczące implementacji systemu operacyjnego.

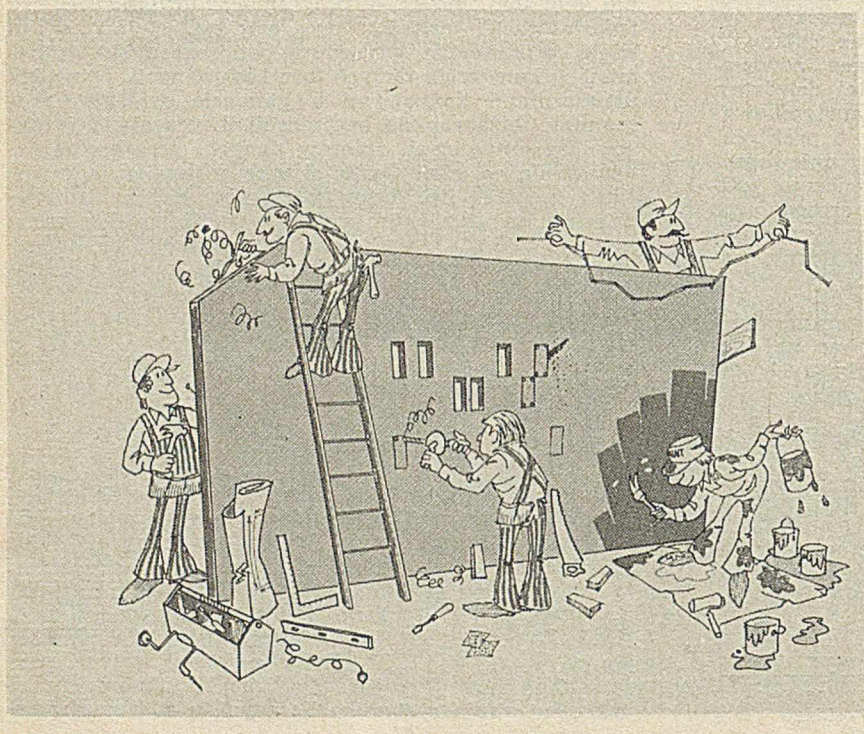
W praktyce, dla badań jakości i porównywania systemów liczących, a tym bardziej projektów ich systemów operacyjnych, wykorzystuje się metody symulacji jako jedyne narzędzie pozwalające uzyskać spojrzenie całościowe; ewentualne korekty polegają zaś na wprowadzaniu zmian do systemu operacyjnego, czy też jego projektu, drogą prób i błędów.

Oczywiście wymienione tutaj trzy główne problemy w dziedzinie systemów operacyjnych nie powinny być traktowane oddzielnie. Wszystkie one są ze sobą ściśle powiązane, tak że praktyczne rozwiązanie jednego z nich będzie mieć wpływ, a nawet może oznaczać rozwiązanie innych problemów w pozostałych kierunkach.

Banki danych

Ten kierunek badań nabiera coraz większego znaczenia, bowiem pojęcie banku danych, niegdyś wiązane przede wszystkim z systemami wyszukiwania informacji, dziś pojawia się niemal we wszystkich typach systemów przetwarzania, np. w systemach obliczeń inżynierskich, systemach zarządzania, planowania itd. Zakres problematyki badawczej związanej z pojęciem banku danych jest bardzo rozległy. Badania w tym zakresie prowadzone w wielu ośrodkach na świecie można podzielić na następujące grupy zagadnień:

- konstrukcja baz danych
- zagadnienia formalizacji opisu struktur danych i języki opisu danych
- metody i środki programowe dla definiowania zbioru operacji na bazie danych
- zagadnienie komunikacji użytkownik — bank danych.



Rozważmy pokrótce powyższe zagadnienia.

Konstrukcja bazy danych

Szybki rozwój informatyki stale zwiększa wymagania w zakresie możliwości przechowywania i przetwarzania coraz większych zbiorów informacji. Podstawowym zagadnieniem jest tu więc konstrukcja baz danych, tj. zbiorów informacji spełniających określone cechy. Do cech tych należy przede wszystkim zaliczyć:

— niepowtrzalność danych czyli eliminowanie redundancji, dzięki czemu nie tylko zmniejsza się wymagania na pojemność nośników informacji, ale ułatwia się aktualizację zbiorów i unika się sprzeczności wynikających z niejednoczesnej aktualizacji wielokrotnie zapisanych informacji.

— zapewnienie ochrony danych, umożliwiające dostęp do danych tylko upoważnionym użytkownikom.

W chwili obecnej istnieje wiele kierunków badań związanych z konstrukcją baz danych. Z jednej strony prowadzi się próby interpretacji, istniejących sformalizowanych teorii matematycznych dla potrzeb konstrukcji baz danych, z drugiej zaś strony próbuje się tworzyć nowe modele, jak np. modele relacyjne baz danych. Ponadto opracowano w tym zakresie propozycję standardu (prace prowadzone przez grupę CODASYL), jednakże z teoretycznego punktu widzenia propozycja ta nie jest dostatecznie uzasadniona.

Mówiąc o konstrukcji baz danych należy szczególną uwagę zwrócić na tzw. bazy rozproszone, tj. takie, które są podzielone na części przechowywane w odległych od siebie miejscach. Dostęp do wszystkich części każdej bazy jest zapewniony przez sieć transmisji danych oraz wysoce rozwinięte oprogramowanie specjalne.

Użytkownik takiej bazy nie orientuje się nawet, z której części bazy korzysta.

Opis struktur danych

Niezmiernie ważnym zagadnieniem związanym z konstrukcją baz danych jest jednolita klasyfikacja struktur danych oraz język do opisu tych struktur. Istnieje pilna potrzeba opracowania takiego języka, jako narzędzia użytkownika umożliwiającego mu definiowanie dowolnych struktur. Obecnie możliwości te są w istotny sposób ograniczone bądź przez istniejące języki programowania, bądź też przez pracochłonność opisu tych struktur.

Operacje na zbiorach i metody dostępu

Jest rzeczą oczywistą, że przechowywanie informacji jest tylko środkiem niezbędnym do osiągnięcia przez użytkownika określonego celu. Celem bezpośrednim, przynajmniej w sensie przetwarzania informacji jest wykonywanie określonych operacji na bazie danych. Jak się okazuje, intuicyjnie można wyróżnić wiele operacji, które często występują. Konieczne jest jednak przeprowadzenie szczegółowych badań prowadzących do jednolitej klasyfikacji zbioru wszystkich operacji, a następnie do wybrania takiej klasy operacji, na podstawie której można byłoby generować dowolne inne operacje.

Wykonywanie operacji jest ściśle związane z metodami dostępu do zbiorów. Potrzebne są więc badania metod dostępu i ich wpływu na sposób wykonywania poszczególnych operacji, przy czym pod uwagę należy brać efektywność metod dostępu w zależności od organizacji bazy danych.

Komunikacja użytkownik — baza danych

Obecnie wraz z rozwojem zastosowań oraz wzrostem stopnia ich skomplikowania coraz większą wagę przywiązuje się do sposobu formułowania żądań przez użytkownika. Stosowane dotychczas formy — pisanie programów nie mogą być w dalszym ciągu akceptowane. Pierwsze prace w tym kierunku objęły systemy wyszukiwania informacji. W wyniku tych prac użytkownik może formułować swoje wymagania, tj. kryteria wyszukiwania informacji w sposób zbliżony do tradycyjnego toku myślenia. Badania powinny wykazać, w jaki sposób oraz w jakim zakresie operacji winny być w pierwszym rzędzie udoskonalone formy komunikacji człowiek — maszyna.

Translatory

Prace w tej dziedzinie prowadzone są w Polsce najdłużej, największe jest też zgromadzone doświadczenie. Uzasadnienie celowości tych prac jest doskonale znane, dlatego nie przytaczamy go tutaj.

Aktualnie najważniejsze prace w zakresie konstrukcji translatorów są prowadzone w dwóch kierunkach.

- Tworzenie języków maszynowo zorientowanych (np. BLISS), dostatecznie wysokiego rzędu, umożliwiających opis procesu translacji, na ogół sterowanego składnią.

Języki te mają szereg zalet:

a) umożliwiają produkcję optymalnych translatorów
b) umożliwiają strukturalne konstruowanie translatorów
c) dają precyzyjną dokumentację prowadzonych prac.

Wadą tego rozwiązania jest to, że konstruktor musi nie tylko znać język opisujący, lecz również metody translacji.

- Tworzenie metod formalnego opisu języków, ich składni i semantyki oraz definiowanie systemów, które na podstawie opisu języka automatycznie konstruują jego translator.

Pośród prac prowadzonych w ramach drugiego kierunku można wydzielić następujące grupy tematyczne.

Prace związane z warstwą syntaktyczną języków. Do grupy tej zaliczamy prace nad metodami opisu składni oraz metodami analizy składni; do tych ostatnich zaliczamy również prace nad analizą leksykalną. Jak dotąd istotną rolę w omawianej grupie zagadnień odgrywały badania nad językami bezkontekstowymi. Najbardziej popularną metodą opisu składni języka jest tzw. notacja BNF oraz jej weryfikacje; inne metody (jak np. Język produkcji Floyda) nie znalazły szerszego zastosowania. Jeżeli chodzi o metody analizy składni, to opracowano szereg takich metod, jak np. metody z operatorowym, prostym, słabym i uogólnionym pierwszeństwem, metody z ograniczonym i mieszanym pierwszeństwem, metoda LR(k) i jej modyfikacje, metoda LL(k), itd. Obecnie, przede wszystkim dzięki wprowadzeniu gramatyk dwupoziomowych (gramatyki van Wijngaardena, gramatyki afiksowe) główny akcent przenosi się na badania w zakresie języków kontekstowych. Problemy, które należy tutaj rozwiązać, to opracowanie efektywnych metod analizy składni tych języków, jak również zastosowania ww. gramatyk do opisu tzw. semantyki statycznej języków.

Prace w zakresie semantyki. Są one prowadzone w dwu kierunkach: teoretycznym i realizacyjnym. „Teoretycy" formalizują pojęcie programu w terminach algebry relacji lub funkcji bądź definiują język za pomocą zbioru aksjomatów i dołączonych do nich reguł wywodu, bądź wreszcie budują model języka w oparciu o aparat obiektów abstrakcyjnych. „Praktycy" z kolei proponują różne języki dla definiowania akcji semantycznych sterowanych składnią, bądź definiują atrybuty stałych i zmiennych gramatyki opisywanego języka. Jak się wydaje wyniki teoretyków (z wyjątkiem, być może, prac prowadzonych przez Scott'a i Strachey'a) nie osiągnęły jeszcze takiego stadium, aby można je było stosować w praktyce, zaś propozycje praktyków są za bardzo pragmatyczne, zbytnio obciążone zarówno konkretnymi metodami translacji, jak i architekturą maszyn, na których te propozycje są realizowane. Dotyczy to szczególnie semantyki dynamicznej.

Jak dotąd wyniki prac w tym zakresie nie są zadowalające. Znane dotychczas metody opisu języków nie gwarantują otrzymania optymalnych translatorów tych języków; istniejące metajęzyki są skomplikowane zwłaszcza w części dotyczącej semantyki i pragmatyki.

Wydaje się, że pierwsze z tych rozwiązań jest bardziej dogodne dla konstruowania translatorów dużych, uniwersalnych języków — programowania. Natomiast drugie podejście może być dogodne przy konstruowaniu translatorów dla specjalistycznych języków zastosowaniowych.

Warto zwrócić uwagę, że tematyka związana z translatorami obejmuje także zagadnienia mikroprogramowania maszyn cyfrowych.

Komunikacja człowiek — maszyna

Ten kierunek badań dotyczy nowych technicznych możliwości komunikacji człowiek — maszyna, a szczególnie tych, które stwarzają urządzenia komunikacji graficznej. Możliwości te są niezwykle istotne dla wielu dziedzin zastosowań informatyki. Wprowadzają

one nowy język wymiany informacji między człowiekiem i maszyną, operujący obiektami wielowymiarowymi. Stąd wynika szereg problemów badawczych, których rozwiązanie winno przynieść w szczególności metody i środki programowe dla identyfikacji i generacji obrazów oraz operowania nimi w maszynie. Tematyka ta niewątpliwie wiąże się z badaniami z zakresu teorii struktur danych.
Obok wymienionych czterech kierunków przedmiotowych, do głównych kierunków badawczych zaliczyć należy również niżej wymienione zagadnienia uniwersalne, dotyczące oprogramowania jako całości, mimo, że konkretne wyniki odniesione do poszczególnych tematów przedmiotowych będą zapewnie różniły się między sobą.

Technologia wytwarzania oprogramowania

Wśród zagadnień szczegółowych tej dziedziny, na pierwszy plan wysuwają się:

- Stworzenie praktycznie stosowalnej metodologii projektowania oprogramowania. Obiecujące w tym zakresie wydają się zalecenia inżynierii oprogramowania, w sensie zdefiniowanym przez konferencje w Garmisch-Partenkirchen i Rzymie, a w szczególności projektowanie zastępujące (analityczne), modularna morfologia, pierwszeństwo poprawności i niezawodności względem optymalności.
- Stworzenie zespołu środków wspomagających projektowanie i wytwarzanie oprogramowania przez rozwinięcie zasad i praktyczne wdrożenie języków funkcjonalnego opisu oprogramowania, języków częściowej specyfikacji oprogramowania oraz kompilatorów krzyżowych.

Metodyka weryfikacji oprogramowania

W dziedzinie tej wyodrębnia się obecnie trzy główne kierunki:

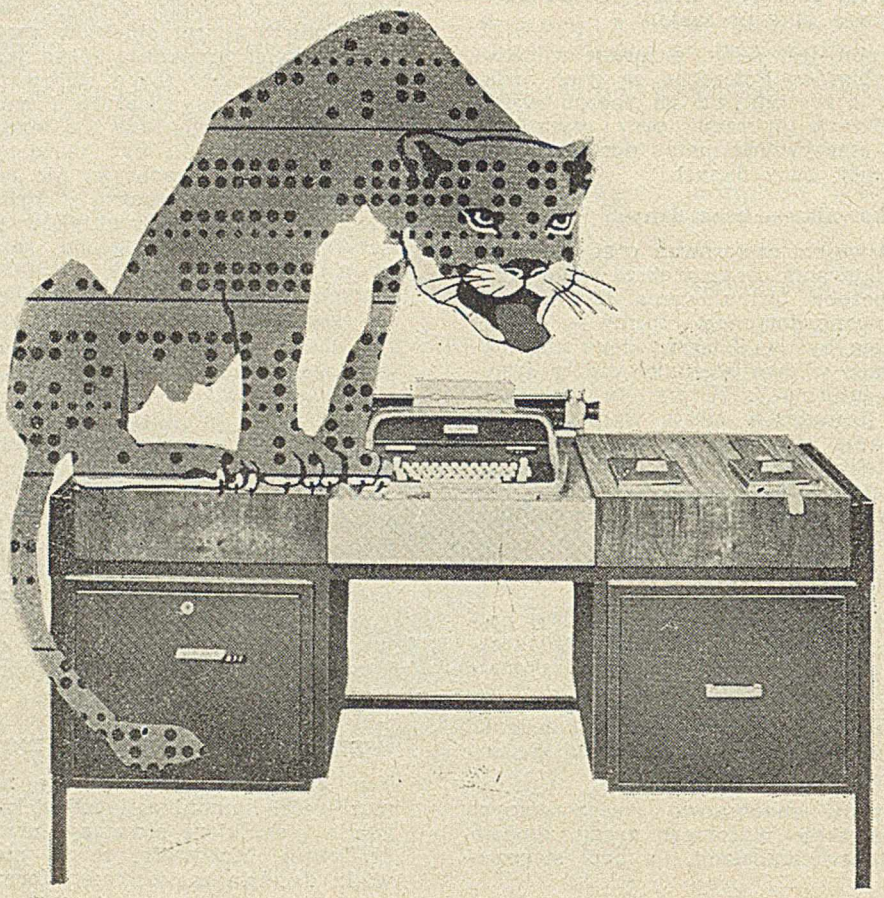
- Metodyka dowodzenia poprawności programów, opierająca się i ściśle związana z formalną definicją semantyki języków programowania.

- Metodyka testowania i atestowania oprogramowania wiążąca się z implementacją oprogramowania i architekturą systemów liczących. Testowanie dotyczy badania poprawności implementacji poprawnego oprogramowania w ogóle, zaś atestowanie — badania przydatności poprawnie implementowanego poprawnego oprogramowania.

- Metodyka określania stabilności oprogramowania, dotycząca oceny zachowania się oprogramowania przy małych odchyleniach od normy

a) danych
b) działania lub konfiguracji sprzętu

W wykazie najważniejszych kierunków badawczych pominęliśmy kilka tematów tradycyjnie wymienianych w poprzednich materiałach podobnego charakteru, jak również należące do problematyki sprzętowej zagadnienia automatyzacji projektowania maszyn cyfrowych, zawierające szereg tematów badawczych z dziedziny oprogramowania. W szczególności pominęliśmy: konstrukcję nowych języków programowania, nowych pakietów programów, oprogramowanie nowych typów urządzeń cyfrowych, budowę typowych systemów przetwarzania informacji. Uznając znaczną wagę praktyczną takich prac, a nawet — jak w przypadku typowych systemów przetwarzania informacji do celów zarządzania — ich kluczową rolę w uzyskaniu tak pożądanych sukcesów informatyki, uważamy, że w tych kierunkach osiągnięto już wystarczający poziom wiedzy, pozwalający na prowadzenie prac realizacyjnych bez intensywnych badań naukowych (w każdym razie w zakresie oprogramowania).



ZYGMUNT RYZNAR
Ośrodek Organizacji i Informatyki
Przemysłu Petrochemicznego PETROINFORM
Kraków

681.327.2.004.141:651.8:002.55:65.012.4

Problemy integracji systemu informatycznego

„Dlaczego nie potrafimy znaleźć rzeczywiście zintegrowanych rozwiązań? Odpowiedź jest nader prosta. Jest tak dlatego, ponieważ istnieje znacznie więcej opowiadania o sukcesach, niż samych sukcesów. Więcej daje się przyrzeczeń, niż się ich dotrzymuje, a ponieważ przyrzeka się zbyt wiele i za wcześnie, istnieje określony stopień frustracji, która staje się coraz bardziej powszechna".

(Adrian M. McDonough)

Autor podkreśla znaczenie integracji danych dla elastycznego wykorzystywania systemów informatycznych w zarządzaniu oraz dla racjonalizacji procesów przetwarzania, programowania i utrzymania systemów. Zaznacza, że wspólne zbiory danych powinny być projektowane z uwzględnieniem dalszego rozwoju systemów. Omawia treść integracji i metody jej wprowadzania.

Czy istnieje rzeczywista potrzeba integracji?

Informatykę lat 60-tych charakteryzowało pojawienie się swoistej „ideologi" pojęć, lansowanych niekiedy wbrew rzeczywistym osiągnięciom i ówczesnym możliwościom. Do pojęć tego typu zaliczyć można określenie „zintegrowany system informowania kierownictwa".

Od optymizmu nie było wolne również opracowanie Europejskiego Programu Badawczego Diebolda [1]. Pojęcie IMIS *(Integrated Management Information System)* zdefiniowano tam w następujący sposób: „jest to system informacyjny zbudowany z wykorzystaniem wszelkich dostępnych środków w celu wyposażenia kierowników wszystkich szczebli i w zakresie wszystkich funkcji w informacje ze wszystkich odpowiednich źródeł w celu umożliwienia im terminowego i skutecznego podejmowania decyzji w zakresie planowania, kierowania i nadzorowania czynności, za które są odpowiedzialni".

Definicja powyższa, przy całej swojej słuszności, odnosi się do systemu budowanego w optymalnych warunkach bez istotnych ograniczeń.

Naturalnym biegiem rzeczy, wobec małej konkretności pojęcia „system zintegrowany", zaczęto stosować określenie „systemy oparte na bazie danych" *(data base information systems)* lub też inne terminy. Czy jednak idea systemów zintegrowanych była tylko przejawem mody i obecne koncepcje całkowicie ją eliminują? Czy jej werbalizm nie wynikał między innymi z tego, że była przedwczesna w stosunku do osiągniętego poziomu zastosowań?

Złożoność ujęcia projektowego dostosowana być bowiem powinna do poziomu systemu, który zamierzamy projektować. Poziom ten decyduje o treści i użyteczności systemu. Z punktu widzenia walorów użytkowych wyróżnić można następujące poziomy systemów:

- poziom ewidencyjno-rozliczeniowy
- poziom informacyjno-kontrolny (np. kontrola zużycia materiałów)
- poziom informacyjno-decyzyjny (wymagający zastosowania modelu postępowania, dostępu do różnorodnych informacji w praktycznie dowolnych momentach czasu, itp.).

Jeśli granice systemu zamykają się w poziomie ewidencyjno-rozliczeniowym, wówczas stosowanie skomplikowanych rozwiązań projektowo-technologicznych (np. banku danych użytkowanego w trybie pytanie—odpowiedź) uznać można bardziej za rezultat ambicji, niż rzeczywistych potrzeb. Jeśli natomiast poziomy wynikają z siebie nazwajem, w szczególności zaś poziom informacyjno-decyzyjny opiera się na informacjach zgromadzonych na poziomie najniższym, wtedy należy myśleć o wprowadzeniu wspólnej bazy danych i zapewnić odpowiedni sposób jej zasilania. Poziom najwyższy wymaga bowiem scalenia (z punktu widzenia dostępu) wielu różnorodnych informacji, czego nie można osiągnąć w rozwiązaniach klasycznych. Załóżmy na przykład, że dyrektora ds. handlowych zaniepokoił spadek sprzedaży wyrobu X i w związku z tym zażądał on następujących informacji:

— wielkość sprzedaży w kwartałach roku bieżącego i roku ubiegłego
— poziom zapasów w ostatnich trzech miesiącach
— dane o jakości surowców na przestrzeni ostatniego roku
— liczba reklamacji otrzymanych od odbiorców
— wielkość kosztów własnych, w tym kosztów materiałowych.

W systemie pozbawionym wspólnej bazy danych dostarczenie odpowiedzi na tak złożone zapytanie wymagałoby skierowania kilkunastu prostych pytań do wielu odbiorców w co najmniej kilku przebiegach maszynowych.

W świetle powyższego zrozumiałe są zastrzeżenia wysuwane pod adresem systemów informatycznych, które — mimo swej atrakcyjności — są niekiedy tak sztywne, że znacznie ograniczają swobodę działania kierownictwa, dostarczając jedynie okresowych zestawień w określonej z góry formie. Należy sądzić, że wada ta jest w pewnym stopniu wynikiem stosowania skądinąd słusznej zasady „najpierw porządkować, później automatyzować", czy też „najpierw organizacja, później automatyzacja", prowadząca często do utrwalenia zrutynizowanych algorytmów ewidencyjno-rozliczeniowych wraz z istniejącą strukturą organizacją. Tymczasem elementy te stanowią pochodną systemu zarządzania, który zmienia się wraz ze zmianą sytuacji gospodarczej i doskonaleniem metod zarządzania.

Efektem podejścia tradycyjnego jest konieczność ciągłej modyfikacji systemu (często już na etapie wdrażania), prowadzącej w końcu do stworzenia zupełnie nowego systemu, tyle że większym kosztem.

Należy stąd wyciągnąć wniosek, że **potrzebna jest metodyka projektowania zapewniająca systemowi niezbędny stopień elastyczności,** umożliwiającej łatwe dostosowanie systemu do zmiennych potrzeb systemu zarządzania.

Osiągnąć to można, między innymi, stosując integrację danych, która zostanie bliżej scharakteryzowana w dalszej części artykułu. Niezależnie od znaczenia tej integracji dla samego systemu zarządzania wywiera ona również istotny wpływ na racjonalizację procesu przetwarzania i programowania. Zmniejszenie liczby zbiorów prowadzi do obniżenia kosztów utrzymywania systemu (modyfikacji i aktualizacji zbiorów) oraz do redukcji powtarzania (dublowania) tych samych danych. System dysponujący oprogramowaniem w zakresie zarządzania danymi umożliwia uniezależnienie programów od alokacji i struktur danych, tzn. pozwala programiście działać niejako na poziomie użytkownika a nie na poziomie komputera.

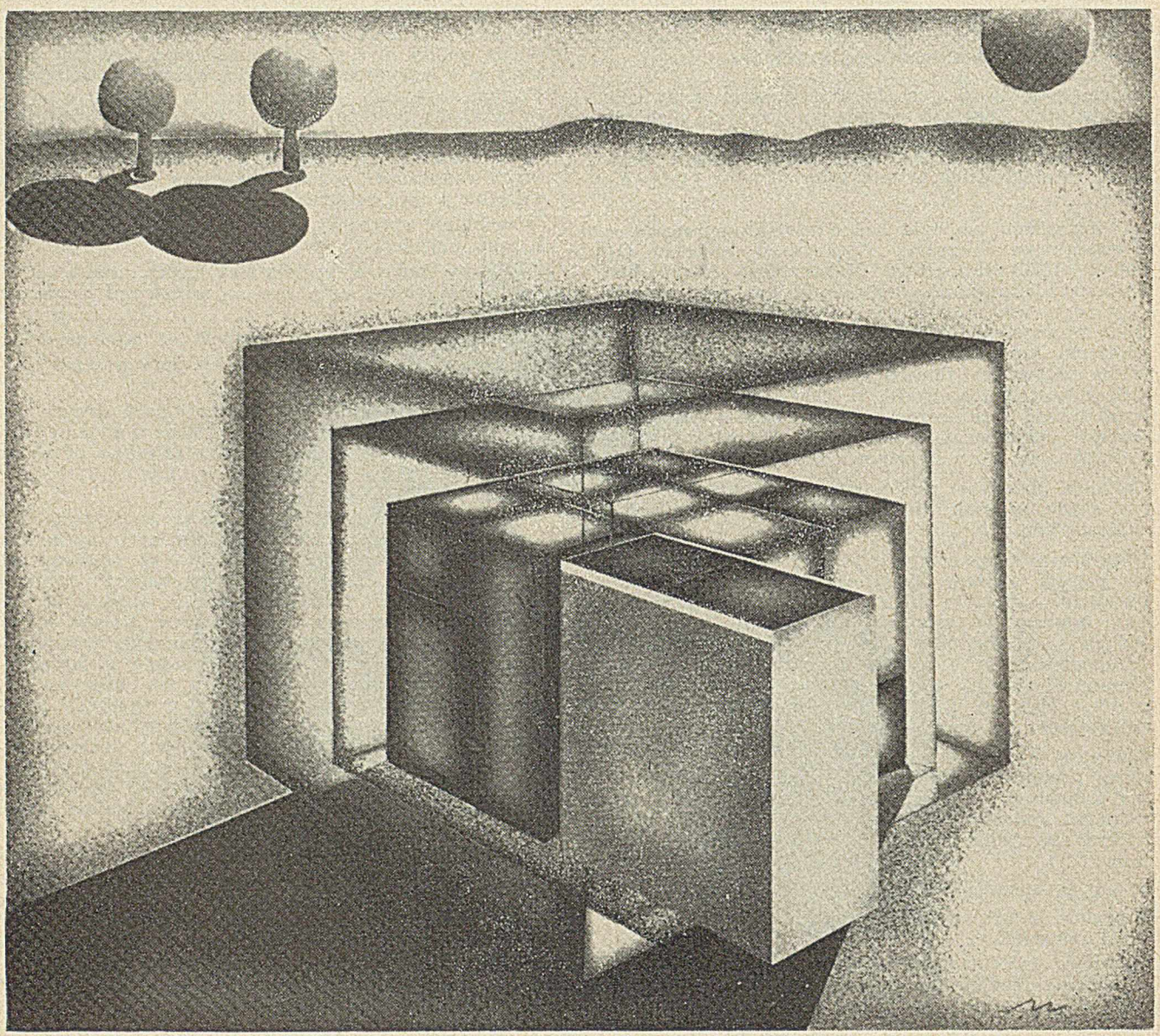
Próba analizy treści integracji

Integracji nie należy mylić z kompleksowością. Kompleksowość jest miarą zakresu i rezultatem działania „wszerz", natomiast integracja jest działaniem „w głąb" i związana jest ze sposobem funkcjonowania systemu. Jeśli integrację systemu określimy jako scalenie jego elementów składowych, będzie to definicja mało przydatna do celów projektowych.

Integrację można realizować w postaci bardzo ogólnej **na poziomie systemu jako całości.** Wyraża się ona wtedy wyznaczeniem celów wiążących poszczególne podsystemy oraz zapewnieniem powiązań informacyjnych. Integracja tak rozumiana jeszcze nie implikuje stosowania specyficznych metod projektowo-technologicznych, lecz toleruje również klasycznie zorganizowane zbiory pod warunkiem, że są one wykorzystywane w kilku podsystemach. Ta forma integracji jest stosunkowo popularna. Integracja w tym ujęciu „jest techniką wielokrotnego użycia danych wejściowych oraz wykorzystania wyjścia z jednego podsystemu jako wejścia do innego" [2].

Główny wysiłek skierowany jest więc na stworzenie jak największej ilości zbiorów wspólnych (wychodzących poza jeden podsystem), które powinny być projektowane z wyczuciem perspektyw rozwojowych systemu.

Zbiory wspólne spełniają szczególną rolę w systemach branżowych, w których obok elementów ogólnobranżowych (takich, jak np. indeksy branżowe) powinny wystąpić również elementy specyficzne dla konkretnych przedsiębiorstw. Przykładem zbioru łącznikowego pomiędzy branżowością, a specyfiką przedsiębiorstwa może być np. „Katalog danych stałych zasobu", zawierający dane dla kilku podsystemów (branżowy symbol zasobu, zakładowy symbol zasobu, symbol sprawozdawczy; nazwy: zakładowa, branżowa, sprawozdawcza, do celów rozliczania kosztów; symbol zasobu stosowany przez biuro zbytu, symbol stanowiska kosztów; jednostki miary: zaopatrzeniowo-handlowa, statystyczna, technologiczna; oceny; ilości planowane, itp.).



Na poziomie technologii wyróżnić można integrację danych i integrację procedur, prowadzące do stworzenia wspólnej bazy danych i wspólnej bazy procedur. Właśnie w tych formach integracji należy szukać przesłanek elastycznego zestawiania danych wynikowych w układach odpowiadających aktualnemu zapotrzebowaniu. Integracja ogólna na poziomie systemu, wyrażająca się tylko w tematycznych powiązaniach zbiorów z pozostawieniem ich klasycznej struktury, nie daje takich możliwości, gdyż ogranicza powiązania pomiędzy danymi wskutek sztywnego charakteru programu napisanego dla ściśle określonej kombinacji zbiorów. Bank danych operujący na danych elementarnych (a nie na zbiorach) jest wolny od tych ograniczeń, wymaga jednak stosowania centralnych identyfikatorów (wspólnego kodu klasyfikacyjnego) jednoznacznie porządkujących elementy wspólnej bazy danych [1]. Banki danych (składające się z bazy danych i systemu zarządzania tą bazą) realizowane być mogą na poziomie systemu operacyjnego, stanowiąc jego integralną część, lub też na poziomie programów użytkowych (np. IDS). Ze względu na różnorodne rozwiązania nie każdy bank danych spełnia całkowicie wymogi integracyjne.

Najistotniejsze cechy integracji na poziomie technologii są następujące:

1) radykalne zmniejszenie stopnia rozproszenia danych po wielu zbiorach (najlepiej poprzez wprowadzenie wspólnej bazy danych)

2) oderwanie sfery użytkowania informacji od miejsc zasilania banku (czyli zlikwidowanie „właścicieli" informacji hamujących naturalny przepływ informacji w przedsiębiorstwie)

3) stosowanie jednolitej symbolizacji i nazewnictwa danych w całym systemie

4) przetwarzanie różnorodnych struktur danych

5) występowanie relacji porządkująco-sprzęgających, określonych dla wspólnej bazy danych, umożliwiających sprawne (efektywne) wybieranie powiązanych ze sobą informacji

6) integracja procedur — upraszczająca strukturę i wielkość programów [2]

7) stosowanie tzw. parametryzowanych agregatów wydawniczych (PAW), eksploatowanych w trybie pytanie—odpowiedź i umożliwiających otrzymanie wielu komunikatów w ramach jednego programu.

Głównym problemem budowy banku danych jest **organizacja wspólnej bazy danych.** Zakładając, że jest to centralny zbiór, mamy na myśli nie fizyczna koncentrację danych lecz ich logiczne (programowe) scalenie. Baza danych może być zlokalizowana nawet w różnych miejscach i w różnych komputerach, sprzężonych liniami teletransmisji [3, s. 335].
O centralnym charakterze zbioru decyduje fakt, że użytkownik żądając określonego zestawu informacji, nie podaje ani lokalizacji jego elementów, ani też istniejących pomiędzy nimi powiązań.

Wspólna baza danych powina być zaprojektowana z uwzględnieniem wszystkich podstawowych funkcji gospodarczych i zarządczych [3] występujących w przedsiębiorstwie czy instytucji. Tak ujęta baza reprezentuje sobą p a s y w n ą strukturę informacyjną systemu, na podstawie której komponowane są moduły użytkowe i podsystemy, przeznaczone do realizacji konkretnych celów i zadań. W tym ujęciu moduły mogą wzajemnie zachodzić na siebie bez dublowania elementów we wspólnej bazie danych. Ułatwia to znacznie projektowanie systemu, zapewniając mu i pewną stabilność i elastyczność oraz zmniejszając nakłady na modernizację systemu.

Podsumowując rozważania nad treścią integracji, można stwierdzić, że może ona obejmować również stosowanie w systemie rónorodnych, wspomagających się wzajemnie środków technicznych. Tendencje integracyjne w ramach systemu informatycznego służą lepszemu ujęciu zjawisk gospodarczych, stanowiących w istocie rzeczy rezultat ciągłego oddziaływania wielu elementów, umownie rejestrowanych w postaci dokumentów, zbiorów, itp. Systemy wyposażone we wspólną bazę danych są środkiem do poznania zjawisk poprzez swobodną analizę wielu różnorodnych informacji zgromadzonych w bazie.

Autor przeanalizował powiązania informacyjne występujące w wielotematycznym systemie informatycznym zaprojektowanym przez jeden z branżowych przemysłowych ośrodków informatyki. W okresie przeprowadzania analizy system ten zawierał 33 zbiory, pomiędzy którymi wystąpiło 1060 powiązań typu powtórzenia elementu. Ponieważ suma elementów w zbiorach wynosiła 1500, oznacza to, że średnio 70% elementów (danych) zostało powtórzonych. Największa częstotliwość powtórzeń odnosiła się do zbiorów indeksowych (typu: indeks materiałów, indeks wyrobów) oraz zbiorów kartotekowych (typu: kartoteka stanów i obrotów materiałowych).

Wśród podsystemów największą liczbę powiązań wykazały podsystemy: obrotu towarowego, rachunku kosztów i planowania, najmniej zaś podsystem kadrowo-płacowy.

Oceniając powyższe wyniki, można stwierdzić, że w warunkach klasycznych systemów informatycznych, projektowanych bez wspólnej bazy danych, rośnie liczba zbiorów i wzrasta stopień powtarzalności informacji.

Powtarzalność ta wynika z potrzeby powiązania zapisów występujących w wielu zbiorach oraz trudności skoordynowania prac prowadzonych przez wiele zespołów roboczych dążących do wprowadzania „własnych" zbiorów. Jest wreszcie pochodną ograniczonej konfiguracji komputera (liczby jednostek oraz pojemności pamięci dyskowych i taśmowych).

Budowa kompleksowych (wielotematycznych) systemów informatycznych wymaga zastosowania precyzyjnej metody projektowania, gdyż w przeciwnym razie w miarę rozbudowy będą one angażowały dla siebie coraz więcej jednostek pamięci masowych i czasu przetwarzania, a coraz mniej będzie pozostawało środków na rzeczywiste użytkowanie nagromadzonych informacji. Efektywność systemów będzie się obniżać, aby w końcu dojść do punktu, gdy ich sprawna eksploatacja będzie niemożliwa (np. w warunkach posiadania 100 zbiorów maszynowych, które trzeba założyć i następnie aktualizować przed każdym użytkowym przetwarzaniem). Problem ma więc swoją wymowę ekonomiczną i dlatego nie od rzeczy będzie stwierdzenie, że od jego rozwiązania zależy w dużej mierze dalsze doskonalenie systemu zarządzania. Sądzimy, że nadszedł czas, aby metodykę projektowania traktować nie jako odpowiedź na pytanie „jak d o k u m e n t o w a ć prace projektowe" lecz „jak p r o j e k t o w a ć system".

Metody integracji

Wprowadzenie integracji wymaga uprzedniego rozwiązania następujących problemów:

- ustalenia sposobu gromadzenia danych
- określenia sposobu zasilania banku danych
- zastosowania elastycznej i równocześnie efektywnej metody wyszukiwania zestawów informacji na żądanie użytkownika.

W zależności od sposobu gromadzenia danych rozróżnia się banki ze wspólną bazą danych, wielobazowe

---

[1] W celu zapewnienia efektywnego przetwarzania centralny zbiór, ze względu na znaczną pojemność, nie może być zorganizowany na zasadzie „worka śmieci". Zgrupowanie danych zmniejsza znacznie liczbę powiązań, których część jest konieczna ze względu na stosowanie jednokrotnego zapisu.

[2] Pierwszym krokiem prowadzącym do integracji procedur jest metoda programowania modularnego.

[3] Patrz art. autora „Projektowanie wspólnej bazy danych dla potrzeb przemysłu" INFORMATYKA 1/74

Tabela 1

Symbol	Nazwa parametru	Wartość domyślna
RD	rodzaj danych (nr i nazwa o dł. 4 znaków)	nie stosuje się
SI	symbol indeksowy zasobu (indywidualny lub grupowy zgodnie z agregacją rzeczową)	nie stosuje się
TD	typ danych	wartościowo
TA	typ agregacji	koniec okresu
OA	okres agregacji	kwartał
RX	typ roku	bieżący

i wyspecjalizowane. Przykładem banku wyspecjalizowanego jest pakiet BOMP *(Bill Of Material Processor)*, oferowany m.in. przez firmy IBM, HONEYWELL, UNIVAC. Pakiet ten składa się z programów obsługujących powiązania łańcuchowe pomiędzy ściśle określonymi czterema zbiorami (wyroby—części, struktura montażowa, operacje, stanowiska pracy). Wydaje się, że w warunkach występowania kilku takich banków można je zamienić w bank wielobazowy, stosując specjalistyczne zbiory łącznikowe, zawierające adresy dyskowe powiązanych zapisów znajdujących się w różnych zbiorach. Metoda ta ma tę zaletę, że umożliwia sukcesywną rozbudowę systemu bez zasadniczych zmian w strukturze istniejących zbiorów (dodane zostają jedynie nieliczne pola „technologiczne"). Jej wadą jest natomiast znaczna maszynochłonność przetwarzania i ograniczenia w dostępie do zbiorów łącznikowych (jest on możliwy jedynie poprzez tzw. zbiór bazowy) oraz utrudniony system kontroli powtarzania tych samych danych w kilku zbiorach. Na szczególną uwagę zasługuje metoda organizacji bazy danych w postaci centralnego zbioru, czyli wspólnej bazy danych. Zalety tego rozwiązania są następujące:

— umożliwia łatwe wyszukanie wszystkich podstawowych informacji dotyczących przedsiębiorstwa i tworzenie nowych powiązań trudnych do przewidzenia na etapie projektowania; pozwala to dostarczać odpowiedzi nawet na bardzo złożone zapytania angażujące w klasycznym systemie wiele zbiorów

— zabezpiecza przed dublowaniem zapisów o tych samych kluczach alokacyjnych (przy organizacji HISAM — *Hierarchical Indexed Sequential Access Method)*

— upraszcza strukturę informacyjną systemu (w stosunku do metody zbiorów klasycznych), gdyż mimo pozornej złożoności samej bazy danych występuje w niej mniej powiązań (szczególnie pseudospójnych)

— narzuca jednolite nazewnictwo danych

— zmniejsza liczbę jednostek pamięci taśmowej w konfiguracji komputera, wymagając w zamian pamięci dyskowej o znacznej pojemności.

Ze względu na konieczność ograniczenia pojemności bazy danych oraz dostosowania jej układu do typowych zapytań, nie może być ona prostą kumulacją danych występujących w zbiorach transakcyjnych, lecz musi zawierać pewne uporządkowane i zagregowane zestawy informacji. Przykładowo, można zbudować bazę danych wyróżniając w niej poziomy agregacji czasowej, rzeczowej i organizacyjnej. Treść takich poziomów powinna być dostosowana do specyfiki użytkownika. Dla przedsiębiorstwa przemysłowego może być ona następująca:

— agregacja rzeczowa:

0 fizyczna jednostka zasobu
1 reprezentant grupy
2 przeciętny statystyczny reprezentant
3 podzbiór (np. surowce)

— agregacja organizacyjna:

0 gniazdo produkcyjne lub instalacja (oddział)
1 wydział
2 rodzaj działalności w ramach przedsiębiorstwa
3 przedsiębiorstwo razem

— agregacja czasowa:

TA typ agregacji
K na koniec okresu
N narastająco od początku roku
P przeciętnie
OA okres agregacji
D dekadowo
M miesięcznie
K kwartalnie
R rocznie
W wieloletnio
RX typ roku
B bieżący
P poprzedni.

Struktura wspólnej bazy danych powinna być na tyle elastyczna, aby umożliwiać rozbudowę rekordów poprzez dołączanie elementów dodatkowych. Równocześnie musi być prosta, aby wyeliminować zbyt dużą liczbę relacji pomiędzy elementami i zastąpić ją stosunkowo nielicznymi relacjami pomiędzy grupami elementów.

Celowi temu służy poniższy **podział hierarchiczny elementów:**

— podzbiór (największa jednostka danych w ramach wspólnej bazy danych)
— zapis fizyczny (odpowiada blokowi lub stringowi)
— segment (zapis logiczny w konwencji języka COBOL)
— sektor (jako pole grupowe)
— dane elementarne.

Łańcuchowe struktury powinny być wprowadzane na poziomie podzbioru, zapisu i segmentu. Łańcuchy na poziomie podzbioru służą np. do łączenia zapisów dotyczących materiałów z zapisami dotyczącymi dostawców i zamówień, co pozwala na znaczne skrócenie długości łańcuchów. Łańcuchy na poziomie zapisów umożliwiają grupowanie zapisów w podzbiory. Łańcuchy segmentów są niezbędne do zaznaczania powiązań z elementami dodatkowymi lokowanymi np. na zasadzie nadmiaru. Dalsze uszczegółowanie łańcuchów jest z reguły niecelowe, gdyż prowadzi do nadmiernego wzrostu maszynochłonności przetwarzania [4]. Dostęp do (wybranych) danych elementarnych powinien być realizowany raczej poprzez skorowidze adresowe. Wyróżnienie sektorów może być stosowane do wydzielania grup informacji szczególnie ważnych (np. sektor informacji odchyleniowych), generowanych automatycznie dla potrzeb użytkownika przez system.

Najistotniejszym elementem koncepcyjnym **sposobu zasilania wspólnej bazy danych** jest użycie takiej metody symbolizacji, która umożliwiałaby automatyczną konwersję symbolu zdarzenia informacyjnego na adres logiczny (zaś w przypadku zapytania — konwersję na adres procedury czy programu).

Metoda taka powinna w radykalny sposób uprościć programy rozpoznawania typów zapisów (w warunkach wspólnej bazy danych byłby to problem niebagatelny), zwalniając programistę od konieczności określania wielu operacji porównań.

Wprowadzenie jednolitej (centralnej) symbolizacji zdarzeń informacyjnych [5] racjonalizuje procesy ak-

[4] Wg danych zawartych w publikacji [4] s. 61 wprowadzenie do urządzeń pamięciowych komputera złożonej bazy danych o pojemności 9 mln zapisów w systemie DF-2 zajmie ponad 10 tygodni trzyzmianowej pracy maszyny.

[5] Termin „zdarzenie informacyjne" odnosi się do zarejestrowanych zdarzeń transakcyjnych, planistycznych, prognostycznych, normatywnych, dyspozycyjnych, itp.

tualizacji w przypadku małej aktywności wspólnej bazy danych, ponieważ można wówczas odejść od zasady grupowania transakcji w zbiory i aktualizować bazę bezpośrednio rekordami wczytywanymi w dowolnej kolejności i w jednym przebiegu.

Użytkowanie wspólnej bazy danych wymaga zastosowania metod odmiennych do klasycznego przetwarzania partiowego, w którym uzyskiwane są okresowo tabulogramy ze sztywno określonym układem danych. Odmienność polegać może m.in. na parametryzowaniu programów wydawniczych w celu otrzymania wielu komunikatów w ramach jednego programu (programy takie powinny spełniać następujące wymogi:

- gwarantować kontrolę dostępu do danych (poprzez hasło systemu, hasło użytkownika, itp.
- zapewniać jednokierunkowy przepływ danych (na poziomie zapytania użytkownik nie ma prawa wprowadzać żadnych danych do zbioru)
- umożliwiać pracę w trybie konwersacyjnym (dialogu użytkownika z systemem) bez stosowania zwrotów programowych
- umożliwiać przetwarzanie na bieżąco (użytkownik otrzymuje odpowiedź na zapytanie z opóźnieniem wyznaczonym przez czas odpowiedzi tzw. *response time).*

W celu odciążenia użytkownika stosowana powinna być parametryzacja domyślna (w przypadku ominięcia parametru w zapytaniu). Oto przykład zestawu parametrów:

O elastyczności metody parametryzowanych agregatów wydawniczych świadczy fakt, że wg powyższego zestawu parametrów uzyskać można ok. 300 komunikatów.

Uwagi końcowe

Znaczenie banku danych dla usprawnienia systemu informacyjnego oceniane może być bardzo wysoko. Na przykład, Weinmeister [5] wyraża pogląd, że „niepowodzenia systemów informowania kierownictwa, kierowania i sterowania wynikają z niedoceniania problemu konstrukcji bazy danych".

Usprawniający wpływ banku danych określić można następująco:

- narzuca jednolite nazewnictwo (symbolizację) zdarzeń w całym systemie, sprzyjając tym samym wzajemnej komunikacji
- zmniejsza rolę zestawień okresowych, wysuwając na plan pierwszy eksploatację wspólnej bazy danych w trybie pytanie—odpowiedź
- czyni bezzasadnym powtarzanie rejestracji danych w poszczególnych podsystemach
- spełnia rolę buforu synchronizującego różne rytmy obiegu i zapotrzebowań na informacje (np. w przypadku ewidencji operatywnej i księgowej)
- wpływa na skrócenie obiegu danych źródłowych poprzez kierowanie danych, bezpośrednio po ujęciu, do wspólnej bazy danych
- racjonalizuje proces przetwarzania danych.

Wprowadzenie banku danych stwarza więc podstawy do uelastycznienia i uaktywnienia systemu, dając możność bezpośredniego dostępu do informacji wszystkim zainteresowanym (i uprawnionym) decydentom w przedsiębiorstwie.

Literatura

[1] Diebold European Ressarch Program "An executive guide to IMIS" Doc. 31.

[2] Basic Training in Systems Analysis. NCC and Ptiman. 1969 s .45.

[3] P. P. Uhrowczik. Data Dicitionary/Directories. IBM Systems Journal. Vol. 12, No. 4. 1973 s. 332—350.

[4] Europejski Program Badawczy Diebolda. Programy operowania danymi, zesz. 31 (E61) OBRI Warszawa 1973.

[5] Weinmeister C .C. The science of information management. Computer and Automation 4/71, s. 21—24.

---

dok. ze str. 8

chodowego. Efekty ekonomiczne wdrażania ASU w kolejnictwie Białorusi szacuje się na ok. 1 mln rb. Zaawansowane są prace nad projektowaniem i wdrażaniem ASU w wielu instytucjach i przedsiębiorstwach.

Jednakże kontrola przeprowadzona przez władze republikańskie wykazała sporo opóźnień w stosunku do terminów planowych, jak również słabość merytoryczną niektórych projektów.

Prace nad wielkimi systemami ogólnopaństwowymi

Wśród systemów ogólnopaństwowych wielką skalą zagadnień do rozwiązania wyróżnia się zautomatyzowany system statystyki państwowej (ASGS). Do 1975 r. ma być wdrożonych 65 zespołów elektronicznego przetwarzania danych, co umożliwi obliczanie tym sposobem ok. 70% prac statystycznych. Przewiduje się w tym systemie stworzenie zautomatyzowanego banku danych przeznaczonego do wykorzystywania i w innych systemach. Jako jedno z ważniejszych funkcjonalnych ogniw ogólnopaństwowego systemu zarządzania OGAS, system statystyki państwowej musi ściśle współdziałać z innymi systemami, a szczególnie z zautomatyzowanym systemem planowania — ASPR.

Prof. W. Głuszkow, jeden z pionierów informatyki w ZSRR, widzi jako jedno z najważniejszych zagadnień budowy państwowego zautomatyzowanego systemu planowania — analizę i korygowanie bazy normatywnej na wszystkich szczeblach. Obecnie uśredniane wskaźniki, szczególnie na wyższych szczeblach zarządzania, nie odzwierciedlają obiektywnie istniejących rezerw i zapasów, nie uwzględniają postępu technicznego. Prof. Głuszkow proponuje oparcie zautomatyzowanego systemu planowania na czterech modelach, uzupełnianych i korygowanych w sposób nieprzerwany w Gosplanie ZSRR, na podstawie danych spływających z resortów:

- model prognozowania postępu naukowo-technicznego da podstawę do opracowywania planów B + R
- model planowania perspektywicznego ma za zadanie opracowanie, w ramach określonej części dochodu narodowego, bilansu nakładów na wszystkie programy budowy i rekonstrukcji gałęzi gospodarki narodowej
- model planowania długoterminowego ma na celu zapewnienie właściwego wyboru projektów, alokacji i nakładów dla wszystkich inwestycji
- model planowania bieżącego ma zapewnić zsynchronizowanie na poziomie przedsiębiorstw i resortów planów wzajemnych dostaw w ramach specjalnego międzyresortowego systemu obliczeniowego.

Oczywiście, tego rodzaju prace są zakrojone na wiele lat i wymagają ogromnej pomocy ze strony nauki. Problemy te są przedmiotem intensywnych badań w placówkach naukowych ZSRR.

STANISŁAW MATWIN
Pracownia Metod Programowania CO PAN
Warszawa

681.322.06

Język programowania PASCAL i jego kompilacja

Przedstawiono najważniejsze cechy języka PASCAL i jego kompilatorów na komputery CDC 6000.

Język programowania PASCAL [1], którego autorem jest prof. Niklaus Wirth z ETH w Zurichu, powstał na przełomie lat 1968—69. Jego twórcy przyświecały dwa pozornie odległe cele: stworzenie języka będącego podstawą do nauczania programowania w sposób systematyczny oraz zawarcie w nim takiego aparatu, który umożliwiłby użycie PASCAL-a jako narzędzia do budowy tzw. dużych programów.

Obecnie istnieją 2 kompilatory PASCAL-a na maszyny CDC serii 6000; w kilku ośrodkach na świecie trwają prace nad implementacją języka na inne maszyny [2].

PASCAL oparty jest na ALGOL-u 60 z tym, że w bardzo istotny sposób rozszerzono — w porównaniu z ALGOL-em — dostępne w języku struktury danych. Pozwala to na uproszczenie przejścia od zapisu algorytmu w postaci werbalnej lub schematu blokowego do jego wyrażenia w języku programowania.

Struktury danych

Podstawowym pojęciem języka PASCAL jest pojęcie typu. Typem nazywamy pewien zbiór wartości, które może przyjmować zmienna (danego typu).

Typy dzielimy na dwa rodzaje: typy skalarne i typy strukturalne. Typ skalarny określany jest jako uporządkowany zbiór wartości. Podstawą do konstrukcji typów są skalarne typy standardowe: integer, real, Boolean i char [1].

Trzy pierwsze z tych typów należy rozumieć tak samo, jak w ALGOL-u 60, przy czym w typie Boolean **false** poprzedza **ture.**

Typ char oznacza uporządkowany zbiór znaków dostępnych w danej reprezentacji. Oprócz tego programista może definiować własne typy skalarne poprzez wymienienie listy identyfikatorów elementów typu, np.

```
type kolor = (trefl, karo, kier, pik)
     rodzaj = (zwykły, olimpijski, przestępny)
     dzień tygodnia = (pon, wt, śr, czw, pt, sob, niedz)
```

W oparciu o typy skalarne można budować tzw. podtypy, poprzez określenie najmniejszej i największej wartości w podtypie, np.:

```
type młodszy kolor = trefl., karo
     litera = 'A'..'Z'
     cyfra = '0'..'9'
     rok = 0..2000
     X = —200..300
```

W PASCAL-u istnieją 4 podstawowe typy strukturalne: tablice, zapisy, zbiory oraz pliki.

Tablica składa się ze stałej liczby elementów, z których wszystkie są tego samego typu. Liczba wymiarów tablicy nie jest ograniczona; zbiór, przez który mogą przebiegać indeksy danego wymiaru musi być typem skalarnym. Składniki tworzące tablicę mogą być dowolnego typu, np.:

```
type A = array [0..10,'A''Z'] of integer
     B = array [—5..5] of array [2..4] of char
   era = array [rok] of rodzaj
```

Zapis składa się ze stałej liczby składników, które nie muszą być tego samego typu. Każdy taki składnik nazywamy polem zapisu:

```
type student = record nazwisko,
                      imię : alfa;
                      płeć : (mężczyzna,
                              kobieta);
               wzrost, waga : integer;
               średnia ocen : real
          end;
Liczba zespolona = record re, im : real end
```

Algorytmy często działają na obiektach, które — przynajmniej w części opisu — przyjmują różne odmiany. W tej sytuacji zależy nam na możliwości tworzenia różnych wariantów opisu, a więc różnych wariantów zapisu. W PASCAL-u istnieje taka możliwość, ponieważ zapis może mieć tzw. pola zmienne. W tym przypadku wartość pola wskazującego *(tag-field)* określa, który z wariantów pola zmiennego zostanie wybrany.

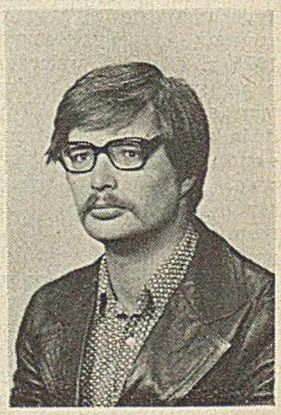
W następującym przykładzie

```
type osoba = record ... lista pól stałych...
             case p : płeć of
                      mężczyzna : (wzrost,
                      waga : integer);
                      kobieta : (wymiary : array
                      [1..3] of integer)
             end
```

[1]) Nieskalarnym typem standardowym jest

```
alfa = packed array 1 .. alfaleng of char
```

gdzie alfaleng jest stałą właściwą danej implementacji, np. w PASCAL-u 6000 alfaleng = 10



Mgr STANISŁAW MATWIN (1949) ukończył Wydział Matematyki i Mechaniki UW, Sekcja Metod Numerycznych (1972). Obecnie jest doktorantem Instytutu Matematycznego PAN. Zajmuje się językami programowania i metodami translacji. Uczestniczy w pracach nad kompilatorem języka PASCAL dla maszyn serii 360/370 w pracowni Metod Programowania Centrum Obliczeniowego PAN.

p jest polem wskazującym typu płeć. O ile w pewnym punkcie programu p ma wartość 'mężczyzna', to znaczy że wybraliśmy w tym miejscu pierwszy wariant zapisu.

Typ zbiór określa rodzinę wszystkich podzbiorów danego typu podstawowego, którym musi być typ skalarny, np.

```
type ręka = set of kolor;
     T = set of 1..10
```

Typ plik jest w PASCAL-u modelem pamięci pomocniczej o dostępie sekwencyjnym; jest to nieskończony ciąg składników tego samego typu. Dla uzyskania dostępu do konkretnego elementu pliku trzeba przejść przez wszystkie poprzednie elementy.

```
type output = file of char;
     list = file of alfa
```

Pozostałe dwa typy, klasa i pointer, są ściśle ze sobą związane.

Klasa pomyślana jest jako jedyna w PASCAL-u dynamiczna struktura danych, tzn. struktura pozwalająca na dynamiczną allokację pamięci.

Klasa opisuje dowolną ilość składników tego samego typu; allokacja pamięci dla elementów klasy następuje w trakcie wykonania programu, w momencie wywołania procedury standardowej „new" od argumentu wskazującego na element klasy. Tym argumentem jest właśnie zmienna typu pointer, która wskazuje element klasy np.:

```
type rodzina = class of osoba ;
     osoba = record          nazwisko : alfa ;
                             ...
                             ojciec, matka : ↑
                             rodzina;
                             ...
             end ;
     var p : ↑ rodzina       (deklaracja zmiennej
                             typu pointer wska-
                             zującej na element
                             klasy rodzina)
begin ...
             new (p) ;       (w tym miejscu w
                             programie następuje
                             allokacja pamięci dla
                             jednego elementu
                             klasy rodzina)
                ...
end
```

Zmienne i wyrażenia

Zmienne w języku PASCAL dzielą się na całkowite i składowe. Zmiennymi całkowitymi nazywamy te zmienne, które oznaczamy przez ich identyfikator. Zmienne składowe dzielą się na: ideksowane, wskazujące pole zapisu i buforowe dla pliku.

Zmienne indeksowane mają budowę podobną, jak zmienne indeksowane w ALGOL-u. Składnik tablicy n-wymiarowej oznaczany jest przez identyfikator tablicy i występujący po nim ciąg n wyrażeń indeksowych, np.:

```
A1 [0, 'C']
B1 [—3] [3]
era 1 [1974]
```

Zmienne wskazujące pole składają się z identyfikatora zapisu do pola którego chcemy się odwołać, kropki i identyfikatora pola, np:

```
student1.płeć
osoba1.p
```

Zmienna buforowa dla pliku opisuje jedyny jego składnik, dostępny w bieżącej pozycji pliku, np.

```
output1↑
list1↑
```

Należy w tym miejscu dodać kilka słów wyjaśnienia o zmiennych typu pointer; jeżeli p jest zmienną typu pointer, to przez p↑ oznaczamy obiekt klasy z którą związany jest pointer, np:

```
osoba1. ojciec↑. nazwisko
```

należy rozumieć następująco: najpierw odwołujemy się do zmiennej (zapisu) osoba1 typu osoba, następnie do pola 'ojciec' tego rekordu, czyli do zmiennej typu osoba wskazanej przez pointer 'ojciec' i wreszcie do pola 'nazwisko' tej zmiennej.

Operatory arytmetyczne w PASCAL-u mają takie same reguły pierwszeństwa jak w ALGOL-u. Rezultat dzielenia jest zawsze typu real. Wprowadzone są operatory dzielenia całkowitego **div** i reszty z dzielenia liczb całkowitych **mod.**

Możliwe są też działania na zmiennych typu set. W tym wypadku użycie operatorów koniunkcji, alternatywy i różnicy arytmetrycznej daje odpowiednio iloczyn, sumę i różnicę zbiorów. Operatory relacyjne $\leqslant, <, \geqslant, >$ mogą działać na zmiennych dowolnych typów skalarnych, ponieważ zgodnie z definicją typy te są uporządkowane. Operatory $\leqslant$, $\geqslant$ działające na zmiennych typu set (zbiorach) pozwalają badać inkluzję zbiorów.

Wprowadzony jest też operator relacyjny **in**, którego drugim argumentem jest wyrażenie typu set, a pierwszym wyrażenie typu podstawowego dla typu set;

```
a in A
```

ma wartość **true** jeżeli a∈A,**false** — w przeciwnym wypadku. Wyrażenia typu set mają postać identyfikatora lub ujętej w nawiasy kwadratowe listy elementów zbioru, z ewentualnym wystąpieniem operatorów:

```
A
[1, 3, 6]
A⋁[1, 3, 6]
```

Instrukcje

Instrukcje języka PASCAL dzielą się na proste i strukturalne. Instrukcje proste to:

- instrukcja podstawienia
- instrukcja **go to**
- instrukcja procedury
- instrukcja pusta

Instrukcja podstawienia jest identyczna jak odpowiednia instrukcja ALGOL-u, z tym ze można wykonywać podstawienia na całych tablicach i zapisach oczywiście tego samego typu. Instrukcja **go to** <etykieta> działa tak samo, jak w innych językach programowania; etykiety są liczbami. Skakać można tylko do etykiet lokalnych lub tych etykiet nielokalnych, które znajdują się na liście definicji etykiet procedury obejmującej.

W PASCAL-u istnieją dwa sposoby komunikowania parametrów aktualnych: przez zmienną (sposób równoważny z fortranowskim 'call by reference' — podawany jest adres parametru) i przez wartość. Parametr formalny może być też procedurą lub funkcją-parametr aktualny jest wtedy komunikowany w sposób analogiczny do algolowskiego wołania przez nazwę.

Deklaracja procedury

procedure A (**var** x : integer; y : integer)

określa, że pierwszy parametr aktualny będzie wołany przez zmienną, a drugi przez wartość.

Instrukcje strukturalne dzielą się na:

instrukcje złożone
instrukcje warunkowe
instrukcje repetycyjne
instrukcje **'with'**

Instrukcja złożona zdefiniowana jest tak samo, jak w ALGOL-u 60. Instrukcje warunkowe dzielą się na **'if — then — else'** oraz **'case'**. Instrukcja **case** jest postaci **case** <wyrażenie> **of** <lista instrukcji z etykietami>. Etykiety muszą być stałymi tego samego typu, co <wyrażenie>, zwane selektorem. Wartość selektora określa, która z instrukcji zostanie wykonana, np. w przykładzie

```
case r of   zwykły: ...
            olimpijski: ...
            przestępny: ...
end
```

selektor r jest wyrażeniem typu 'rodzaj'.

PASCAL zawiera 3 instrukcje repetycyjne:

instrukcję **for**
instrukcję **while**
instrukcję **repeat**

Instrukcja **for** może występować w dwóch postaciach:

for z : = e1 **to** e2 **do** (* e1 < e2 *)

lub

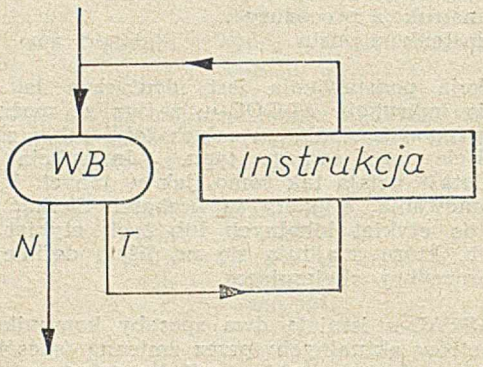
for z: = e1 **downto** e2 **do** (* e1>e2 *)

Zmienna z przyjmuje kolejne wartości typu skalarnego, którego są wyrażenia e1 i e2; jeżeli więc — jak to bywa najczęściej zmienna sterująca jest typu integer to pętla wykonuje się ze stałym krokiem równym 1 lub —1 (w przypadku instrukcji **for** ... **downte** ... **do**)

Instrukcja **while** jest postaci

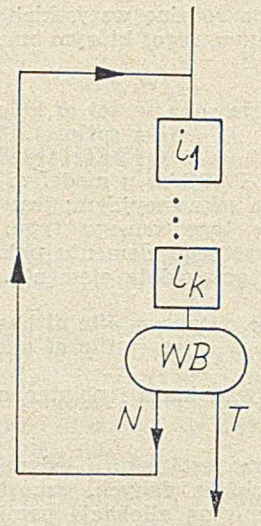
while <wyrażenie boole'owskie> **do** <instrukcja>

i realizuje ona następujący graf sterowania:



natomiast instrukcja **repeat:**
repeat ciąg instrukcji **until** wyrażenie boole'owskie realizuje graf

Instrukcja **with** spełnia w PASCAL-u rolę nawiasu, wewnątrz którego można odwoływać się do pól zapisu bezpośrednio przy pomocy ich identyfikatorów, bez użycia zmiennych wskazujących pole np:




```
with student do if średnia ocen ⩾ 4.5 then
        begin
            wzrost: = wzrost + 4;
             waga: = waga — 5
        end
```

Procedury i funkcje; Program

Deklaracja procedury jest postaci:

<nagłówek procedury> <część deklaracji etykiet> <część definicji stałych> <część definicji typów> <część deklaracji zmiennych> <część deklaracji procedur i funkcji> **begin** <ciąg instrukcji> **end**

Nagłówek procedury składa się z jej nazwy i listy parametrów formalnych; słowa kluczowe **label, const, type, var** rozpoczynają kolejne części deklaracji i definicji. Część definicji etykiet musi zawierać wszystkie etykiety, które występują w instrukcjach **go to** w procedurach lokalnych względem danej procedury i są nielokalne dla tych procedur, np:

```
procedure A; ...
  label 7;
  procedure B;
    begin
      go to 7;
    end (*procedury B*)
begin ... 7: ... end (* proc. A *)
```

Dopuszczalne jest rekursywne wołanie procedur. Oto przykład procedury w PASCAL-u rozwiązującej znane zadanie (zwane wieżami z Benares) [2]) przeniesienia n krążków z jednego stosu na drugi przy pomocy trzeciego, przy czym nie można kłaść większych krążków na zmniejszych:

```
procedure Benares (n:integer; i,j,k: 1...3);
  procedure transfer (s1,s2: 1...3);
    begin (*procedura lokalna realizująca przeniesie-
      nie 1 krążka ze stosu nr s1 na stos nr s2*) ...
    end (* transfer *);
begin
  if n=1 then transfer (i,k)
    else begin Benares (n—1, i,k,j);
               transfer (i,k);
               Benares (n—1,j,i,k)
         end
end.
```

[2]) Zdanie to znane jest też pod nazwą „wieże z Hanoi (przyp., red.).

Funkcje w PASCAL-u mogą być dowolnego typu skalarnego oraz typu pointer, np.

```
function increment (var x : integer): integer;
  begin x: = x + 1 end
```

Program ma w PASCAL-u taką samą składnię, jak procedura; wynika stąd, że w PASCAL-u nie występuje pojęcie bloku w znaczeniu ALGOL-u 60. Warto w tym miejscu zauważyć, że taka konstrukcja programu głównego i procedur narzuca programowanie strukturalną metodą top-down [3]; na wstępie należy określić podstawowe (globalne) struktury danych i te operacje na nich, które przeprowadza się na najwyższym poziomie. Następnie powtarza się to postępowanie na niższym poziomie (procedur kolejnego poziomu) itd. Proces ten jest znany w literaturze jako *step-wise refinement* [4].

Autor języka postuluje umieszczenie w każdej implementacji następujących funkcji i procedur standardowych:
abs, sqrt, sin, cos, exp, ln, arctan, sqr (kwadrat), trunc (argument typu real, funkcja typu integer, trune (x) = entier (x + 0,5)); ord (argument typu char, funkcja typu integer, ord(x) = numer znaku x w danej reprezentacji);
chr — funkcja odwrotna do ord; succ — argument dowolnego typu skalarnego, succ(x) = następnik x, analogicznie pred(x) = poprzednik x (o ile istnieją). Zalecane są także funkcje boole'owskie:

eof(f) = **true** o ile wystąpił koniec pliku f, **false** w przec. wypadku;
odd(x) = **true** o ile x nieparzyste, **false** w przec. wyp.

Ważniejsze procedury standardowe to:

new(p) (p jest pointerem do klasy) powoduje allokację pamięci dla nowego elementu klasy; o ile jest to klasa zapisów z wiariantem, można procedurę new wywoływać od więcej niż jednego parametru. Wołanie new (p,t), gdzie t jest wartością pola wskazującego *(tag-field)* w rekordzie o zmiennym polu, powoduje allokację pamięci dla nowego elementu klasy zapisów z wyborem wariantu odpowiadającego wartości t.

Możliwe jest też zwalnianie w trakcie wykonania programu całej pamięci zajętej dotychczas przez kolejne wołania procedury new.

Do działań na zbiorach dodano w PASCAL-u 6000 różnicę symetryczną (znak operatora '+').

Kompilator PASCAL-a 6000

Kompilator PASCAL-a jest kompilatorem jednoprzebiegowym. Cały system składa się z dwóch zasadniczych części: programu tłumaczącego i pakietu programów zapewniających współpracę z systemem operacyjnym maszyny CDC.

Przy wywołaniu kompilatora można zgłosić włączenie pewnych opcji sprawdzających, które ułatwiają uruchamianie programów. W przypadku przerwania wykonania programu na listingu otrzymuje się adres przerwania oraz wartości zmiennych w momencie przerwania [6].

Program translujący zapisany jest w PASCAL-u; efektem jego działania jest ciąg wołań procedur generujących kod maszyny CDC.

Sam fakt możliwości zapisania translatora języka w samym języku świadczy o jego dobrej przydatności do prac przy oprogramowaniu, o ponadto pozwala łatwo zrozumieć strukturę translatora i sposoby realizacji kolejnych etapów translacji. Możliwe jest także stosunkowo łatwe dokonywanie rozszerzeń języka.

W kompilatorze PASCAL-a 6000 analiza syntaktyczna dokonywana jest metodą diagramów Conway'a, przy czym równocześnie odbywa się generacja kodu. Wybór tej właśnie metody rozbioru wywarł pewien wpływ na szczegóły notacyjne języka (np. zdefiniowanie etykiet jako liczb całkowitych).

Jednoprzebiegowy kompilator generuje kod dla poszczególnych procedur; w czasie wykonania tworzony jest stos segmentów danych dla poszczególnych wołań procedur.

Z procedur standardowych dla plików można wymienić procedury put i get. Procedura put f wpisuje wartość zmiennej buforowej (f↑) do bieżącej pozycji pliku, get(f) wczytuje do f↑ bieżącą pozycję pliku. Procedura reset(f) przewija plik do początku i przygotowuje go do czytania: rewrite(f) przewija f i umożliwia pisanie od początku [5].

PASCAL 6000

Język programowania PASCAL został zrealizowany przez grupę programistów z ETH w Zurichu na maszyny CDC serii 6000 (dwa różne kompilatory). Oprócz tego istnieje jego implementacja na ICL 1900 [9]. Wersja PASCAL-a z listopada 1972 dostępna jest w systemie CYFRONET. Pracownia Metod Programowania CO PAN posiada doświadczenia z ponad półrocznej, intensywnej eksploatacji tego kompilatora, który w dalszym ciągu będziemy nazywać PASCAL-em 6000.

Wersja ta realizuje — z pewnymi wyjątkami — założenia raportu [1] oraz wprowadza szereg rozszerzeń, które pokrótce omówimy. Najważniejszym odstępstwem od zasad raportu jest niedynamiczna realizacja klasy. O ile nie określi się inaczej, przy deklaracji klasy zajmowana jest pamięć dla stu elementów. Jednym z bardziej istotnych rozszerzeń jest wprowadzenie nowej części w programie głównym. Część ta — rozpoczynająca się od słowa **value** — poprzedza część deklaracji procedur i funkcji i służąc do inicjalizacji zmiennych spełnia podobną rolę, jak instrukcja DATA w FORTRA-nie. Oprócz tego instrukcje wejścia wyjścia mają postać znacznie ogólniejszą niż to podano w raporcie.

Założenie jednoprzebiegowości kompilatora przyspiesza kompilację — wobec braku przesłań wyników pośrednich faz translacji między pamięcią operacyjną i pamięciami pomocniczymi. Powoduje ono jednak pewne trudności, związane z realizacją skoków i wołań procedur które poprzedzają ich deklaracje. Przyjętym w kompilatorze PASCAL-a 6000 rozwiązaniem jest przekazywanie z etapu kompilacji do etapu wykonania specjalnej tablicy z adresami skoków.

Projekt realizacji języka PASCAL na maszyny IBM 360/370 i JS RIAD

W licznych ośrodkach badawczych, w tym również i w Polsce, rozpoczęto już prace nad realizacją języka PASCAL na maszyny IBM 360/370, a również i JS EMC.

W jednej z takich prac będzie wykorzystany kompilator PASCAL-a na CDC. Poniżej przedstawiono schemat pracy i użyte programy.

Niech P_b^a oznacza translator PASCAL-a napisany w języku (lub kodzie) a i generujący kod b.

Tak więc:

P_{370}^{PASCAL} — napisany w PASCAL-u translator języka PASCAL z generatorami kodu IBM 370

P_{6000}^{6000} — dostępny w CYRONECIE kompilator PASCAL-a

c.d. na str. 24

Aktualne tendencje rozwoju sprzętu i oprogramowania*)

Nowym, wielkim wydarzeniem w dziedzinie technologii komputerowej jest pojawienie się mikroprocesorów wykonanych w postaci układów o dużym stopniu scalenia. Główni wytwórcy przyrządów półprzewodnikowych rozpoczęli już produkcję mikroprocesorów na taką skalę, że szesnastobitowe urządzenia tego typu, wykonujące rozkaz w czasie 2—3 μs są obecnie dostępne jako mikroukłady kosztujące około 150 dolarów.

Aby uniknąć nieporozumień należy wyjaśnić, że są to poełnowartościowe procesory — „mikro" odnosi się tylko do ich rozmiarów fizycznych i kosztów, a nie do możliwości obliczeniowych. Ich lista rozkazów jest tak samo bogata, jak komputerów z wczesnych lat sześćdziesiątych.

Jedną z bardziej znanych wytwórni produkujących obecnie mikroprocesory jest INTEL Corp. (Santa Clara, Kalifornia). Firma ta wytwarza procesory o słowie zarówno cztero-, jak i ośmiobitowym. Ten drugi typ jest pojedynczym mikroukładem wykonanym w technologii MOS, mieszczącym się w nie przekraczającej długości 1 cala obudowie o 18 wyprowadzeniach. Mniejszy ze wspomnianych procesorów jest wynikiem realizacji programu mającego na celu rozwój kalkulatorów wyposażonych w możliwość drukowania wyników, podczas gdy większy został opracowany z myślą o zastosowaniu go w programowanych urządzeniach końcowych komputerów *(inteligent terminals)*. Inne przedsiębiorstwa, np. NATIONAL SEMICONDUCTOR (Santa Clara, Kalifornia) większy nacisk położyły na modularność mikroprocesorów. Układy wspomnianej firmy, zawierające rejestr i czterobitową jednostkę arytmetyczną (RALU) przystosowane są do łączenia w ośmio-, szesnasto- lub więcej bitowe konfiguracje.

MIKROKOMPUTERY W OBROCIE HANDLOWYM

Mikroprocesory w połączeniu z półprzewodnikowymi pamięciami o dużym stopniu scalenia stają się pełnowartościowymi mikrokomputerami, zdolnymi do podjęcia nie tylko wielu tradycyjnych zadań, ale umożliwiającymi wykorzystanie komputerów w całkowicie nowych dziedzinach.

W ciągu ostatnich kilku lat minikomputery — tanie urządzenia wykonane z obwodów scalonych — stały się dostępne dla użytkowników, których nie stać było na konwencjonalne komputery. Ich wykorzystanie znacznie rozszerzyło obszar zastosowań automatycznego przetwarzania danych. Obecnie, w związku z pojawieniem się mikrokomputerów, powódź nowych zastosowań wdziera się we wszystkie dziedziny życia codziennego. Wszyscy zdążyliśmy już przywyknąć do małych, specjalizowanych komputerów w postaci podręcznych kalkulatorów elektronicznych, lecz mało kto wie o tym, że GM, CHRYSLER i FORD rozpoczęły pertraktacje z producentami przyrządów półprzewodnikowych, mając na uwadze zastosowanie w najbliższej przyszłości „pokładowego" mikrokomputera w każdym samochodzie schodzącym z linii montażowej. Dużo łatwiej dostępne niż ich więksi, droższi poprzednicy, mikrokomputery znajdą się wkrótce w naszych mieszkaniach i urzędach.

Zasadniczo każdy rodzaj działalności mającej charakter powtarzalny może być wspomagany przez mini- i mikrokomputery, które ze względu na niski koszt są dostępne nawet dla najmniejszych firm.

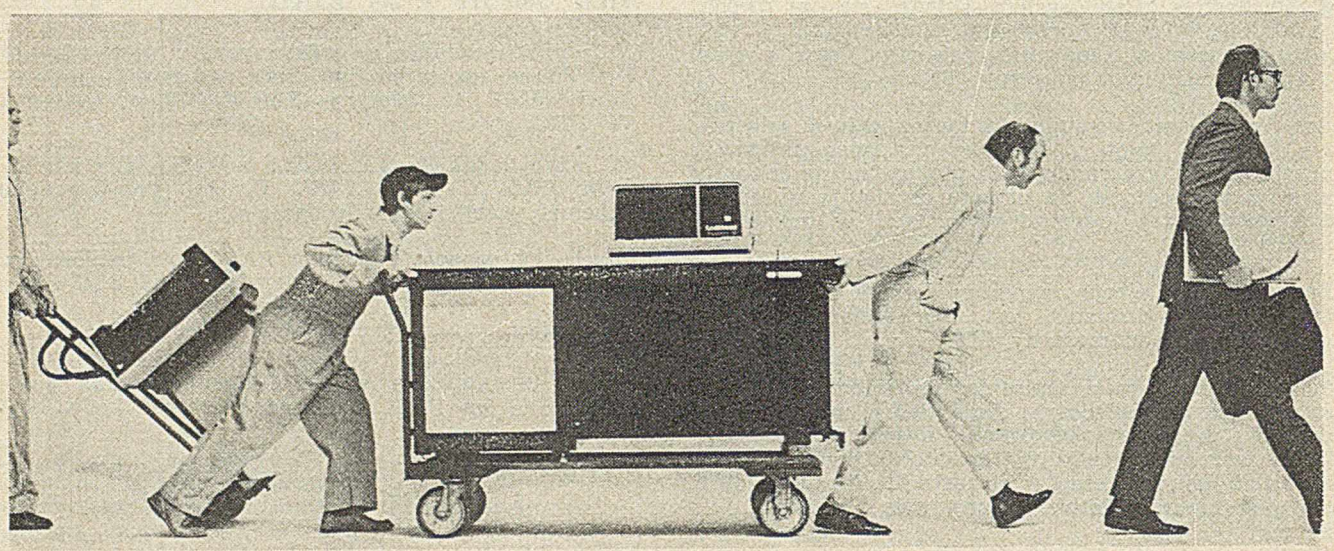
Mikro- i minikomputery niosą także zapowiedź rewolucji w dziedzinie przyrządów pomiarowych. Oczekuje się również, że rychło znajdą zastosowanie w sterowanych numerycznie obrabiarkach, przy automatycznych liniach produkcyjnych, a także w elektronice medycznej, komunikacji i energetyce.

ZAGADNIENIA OCZEKUJĄCE ROZWIĄZANIA

Przedstawione powyżej perspektywy są naprawdę frapujące, lecz pomiędzy obecnym stanem techniki mikrokomputerowej, a jej przyszłym szerokim zastosowaniem leży kilka istotnych problemów, które muszą być rozwiązane.

Pierwszy z nich — to brak użytecznego oprogramowania. Producenci mikroprocesorów dostarczają obecnie translatorów języków typu CROSS-ASSEMBLER,

*) Artykuł opublikowany w IEEE SPECTRUM ze stycznia 1974 r. (s. 39—43): Howard FALK: Technology 74. Computer hardware/software.



które mogą być wykorzystywane w jakimkolwiek komputerze dysponującym FORTRANEM. Dzięki temu program napisany w języku o adresach symbolicznych zostaje przetłumaczony w dużym komputerze na kod wewnętrzny, po czym może być wprowadzony do mikrokomputera i tam wykorzystany. Istnieje także kilka programów symulacyjnych, dzięki którym procedury w kodzie mikrokomputera mogą być generowane w większych komputerach. Wszystkie te sposoby są jednak ograniczone, by sprostać przepływowi zadań stawianych najnowocześniejszym produktom firm wytwarzających układy scalone.

Inne trudności, być może bardziej dotkliwe, dotyczą realizacji współpracy mikroprocesorów z innymi procesorami, z urządzeniami wejścia—wyjścia oraz organizowania dostępu do pamięci wykorzystywanej wspólnie przez kilka mikroprocesorów. Na przykład, nie jest zupełnie jasne, w jaki sposób współczesne mikrokomputery będą dzieliły łączące je magistralę lub komunikowały się z urządzeniami zewnętrznymi. Te problemy z dziedziny problematyki interfejsu mogą być jednak w pewnym momencie przezwyciężone i wtedy zrodzą się nowe perspektywy przed hierarchicznymi, rozproszonymi konfiguracjami sprzętu komputerowego *(distributed systems)*.

Po trzecie, poważnym ograniczeniem zastosowań mikroprocesorów jest ich powolność. Nawet projektowanie kalkulatorów elektronicznych jest poważnie utrudnione przez wielomikrosekundowy czas cyklu logiki MOS — jest on niemalże za długi, by efektywnie obliczać takie elementarne funkcje, jak sinus czy tangens. Jednak w ciągu nadchodzącego roku należy spodziewać się pojawienia się mikroprocesorów o cyklu 1 μs lub nawet krótszym. W przypadku, gdy rozwiązane zostałyby problemy przetwarzania wieloprocesorowego, sprawa prędkości stałaby się mniej istotna.

PAMIĘCI MOS PRZEŁMUJĄ BARIERĘ KOSZTÓW

Jako uzupełnienie mikroprocesorów, produkowane są w technice półprzewodnikowej takie układy pamiętające, jak pamięci MOS o dużej pojemności, szybkie pamięci bipolarne, a nawet jeszcze szybsze układy zawierające diody Schottky'ego.

Jako wynik kilkuletniego gromadzenia doświadczeń wytwórczych nastąpił obecnie zasadniczy przełom w dziedzinie wydajności i kosztów produkcji układów pamiętających MOS. Rezultatem tego jest fakt, że w 1973 r. w wielu gałęziach techniki osiągnięto próg praktycznych zastosowań pamięci półprzewodnikowych.

Mikroukłady pamięci stałej MOS o pojemności 4 Kbitów sprzedawane są obecnie masowo w cenie około 20 dolarów za sztukę. Wobec przewidywanego ciągłego spadku ceny w tempie ok. 20% rocznie, zanosi się na to, że bliski jest czas gdy pamięci rdzeniowe zostaną całkowicie zastąpione przez układy scalone.

KU SYSTEMOM ROZPROSZONYM

Tanie, umożliwiające uzyskiwanie dużego stopnia integracji technologie wykonywania pamięci są — wraz z mikroprocesorami — źródłem ewolucji w zakresie projektowania systemów komputerowych. Jak twierdzi Jack Shemer, redaktor techniczny czasopisma COMPUTER, istnieje wyraźna tendencja zmierzająca również do podziału funkcji komputera. Przeznaczone do realizacji zadanie dzielone jest na mniejsze fragmenty, które przydzielane są do rozwiązania odrębnym częściom mocy obliczeniowej systemu komputerowego .W ten sposób oddzielne procesory wewnątrz komputera mogłyby spełniać takie funkcje, jak przetwarzanie zbioru danych *(file processing)*, sterowanie pamięcią główną, urządzeniami wejścia—wyjścia i operacjami arytmetycznymi.

Taka „rozproszona" architektura systemu nie jest w żadnym wypadku nową koncepcją, lecz dopiero teraz, gdy ma się do dyspozycji wypróbowaną technologię układów MOS wielkiej skali integracji, użycie wielu procesorów w jednym komputerze stanie się prawdopodobnie powszechne.

Kilka firm rozwija obecnie konstrukcję komputerów, w których duża pamięć centralna wykorzystywana jest przez kilka oddzielnych procesorów.

SUPERKOMPUTERY W EKSPLOATACJI

Równolegle z rewolucyjnym rozwojem mikrokomputerów przyniosły także efekty uporczywe wysiłki poświęcone rozwojowi większych komputerów.

Trzy nowe superkomputery weszły do eksploatacji w 1973 r. W Centrum Badawczym NASA rozpoczął pracę w ograniczonym zakresie ILIAC IV. Komputer ASC wyprodukowany przez TEXAS INSTRUMENT dostarczono Agencji Obrony ds. Pocisków Antybalistycznych (Huntsville, Alabama), podczas gdy drugi egzemplarz ACS zainstalowano w Europie. Kolejny superkomputer, STARAN (producent: GOODYEAR AEROSPACE), uruchomiono w Centrum Rozwoju Lotnictwa (Rome, N.Y.).

Istnieje bardzo silna potrzeba postępu w zakresie sprzętu umożliwiającego szybkie operowanie na pamięciach równoległych i przełączanie równoległych strumieni danych. Jak twierdzi David Kuck z Wydziału Informatyki Uniwersytetu Illinois, obecne trudności w realizacji programów równoległych zostałyby w znacznym stopniu zlikwidowane przez takie właśnie udoskonalenia sprzętu. W niedalekiej przyszłości można oczekiwać pojawienia się potężnych superkomputerów o organizacji przepływowej *(pipeline hardware)* [1] i zdolności równoległego operowania dziesiątkami tysięcy bitów. Z pojawieniem się mikroprocesorów możliwość zbudowania maszyny zawierającej 8000 lub więcej procesorów wygląda całkiem realnie.

PAKIETY FUNKCJONALNE PROGRAMÓW OBSŁUGUJĄCE URZĄDZENIA WEJŚCIA — WYJŚCIA

W mini- i mikrokomputerach, w dziedzinie obliczeń równoległych i tworzenia sieci komputerowych, rozwój sprzętu odbywa się w zawrotnym tempie. Niestety, w każdej z tych dziedzin pełne wykorzystanie osiągnięć sprzętowych jest utrudnione przez mniejsze tempo rozwoju prac w zakresie oprogramowania.

Jednym z kluczowych osiągnięć, rozwiązujących po części problemy interfejsu z urządzeniami wejścia—wyjścia dla mini- i mikrokomputerów, jest zastosowanie programów obsługujących urządzenia WE/WY w postaci standardowych pakietów funkcjonalnych *(software kernels)*. W związku z tym użytkownik nie musi się zajmować oprogramowaniem sterującym posiadane urządzenia zewnętrzne. Dokonano także pewnego postępu w rozwoju systemów operacyjnych dla minikomputerów (podobnych do systemów od dawna używanych w większych maszynach). Posuwają się też naprzód, aczkolwiek powoli, prace nad ujednoliconymi językami wyższego rzędu, które obecnie wydają się niezbędne do wielu zastosowań mini- i mikrokomputerów w przemyśle i usługach.

Nie potrafiono jeszcze stworzyć specjalizowanych systemów operacyjnych i języków wyższego rzędu do komunikacji między komputerami. Również, tak istotne dla superkomputerów, oprogramowanie do obliczeń równoległych pozostaje ciągle w początkowej fazie rozwoju.

[1] Organizacja przepływowa — rodzaj organizacji komputera umożliwiającej równoległe przetwarzanie ciągu danych podlegających temu samemu przekształceniu. Operacja dzielona jest na czynności elementarne, które realizowane są w łańcuchu specjalizowanych bloków tak uszeregowanych, aby zapewnić właściwą kolejność obróbki informacji. Każdy z układów po częściowym przetworzeniu informacji przesyła ją do swego sąsiada, gdzie następuje kolejny etap przetwarzania. W ten sposób, kosztem zwiększenia ilości sprzętu odbywają się równolegle w czasie wszystkie operacje elementarne dla określonej liczby danych.

Metoda formularzowa wprowadzania danych (fill the blanks)

W ostatnim roku gwałtownie spadły ceny urządzeń końcowych komputerów, szczególnie urządzeń końcowych programowanych *(inteligent terminals)*, przystosowanych do przetwarzania pewnej klasy informacji. Zrealizowane są one przeważnie w oparciu o mikrokomputery. Takie właśnie urządzenia, dostarczane przez FOUR PHASE SYSTEM (Cuperino, Cal.) pozwalają łatwo wprowadzać dane wprost do komputera, bez użycia kart dziurkowanych. Korzystający z tego sposobu komunikacji są informowani o rodzaju potrzebnych danych poprzez pytania generowane w urządzeniu końcowym i wyświetlane na ekranie monitora. Tworzy to rodzaj kwestionariusza, w którym wolne miejsca wypełniane są przez obsługującego urządzenie.

Programowane urządzenia końcowe, szeroko używane do wprowadzania danych, przydatne są także do przetwarzania informacji w językach naturalnych, szczególnie do redagowania tekstów. Z wbudowanymi procesorami i tanimi, pojemnymi pamięciami, końcówki takie będą wykorzystywane do automatycznego maszynopisania, w składaniu tekstów sterowanym przez komputer, we wstępnej obróbce przy procesie wyszukiwania informacji.

Szeroko rozpowszechnione stało się wykorzystanie programowanych rejestratorów kasowych w punktach sprzedaży detalicznej. W przemyśle spożywczym zaczęto stosować uniwersalny kod oznaczania produktów, który pozwala na użycie w kasach automatycznych optycznego wprowadzania informacji zaszyfrowanej kolorowymi paskami.

DYSKI, ELASTYCZNE DYSKI I GĘSTOŚĆ ZAPISU

Elastyczne dyski *(floppy disks)*, mające wymiary standardowej płyty gramofonowej, lecz wykonane z elastycznego tworzywa sztucznego spowodowały, że cena bardzo pojemnych pamięci o swobodnym dostępie spadła do poziomu warunkującego ich użycie w systemach minikomputerowych. Odkąd dyski stały się wymienialne i wygodne w transporcie, otworzyły drogę do budowy potężniejszych systemów oraz urządzeń końcowych znajdujących się z dala od ośrodka obliczeniowego. Zestawy elastycznych dysków umożliwiające zarejestrowanie 2 milionów bitów i wyprowadzenia informacji szeregowo z szybkością 250 Kzn/s są sprzedawane masowo w cenie nie przekraczającej 800 dolarów. Nawet w porównaniu z jeszcze tańszymi pamięciami kasetowymi, czas swobodnego dostępu elastycznych dysków — mniejszy niż pół sekundy — czyni je bardzo atrakcyjną alternatywą w szerokim zakresie systemów i urządzeń zewnętrznych.

Jedynym ograniczeniem w zastosowaniu dysków elastycznych jest ich mechaniczna wytrzymałość: po około 10 000 dostępów do jakiejkolwiek pozycji, fizyczny kontakt z głowicą zapisująco-odczytującą powoduje starcie powierzchni dysku. Są także produkowane bezkontaktowe dyski elastyczne, lecz to dodatkowe ulepszenie kosztuje jeszcze bardzo drogo.

Dyski konwencjonalne, takie jak IBM 3330, o 200 ścieżkach i gęstości zapisu 4000 bitów na cal — a więc dwukrotnie większej, niż poprzednio — spowodowały odchodzenie od stosowania pamięci taśmowych na rzecz systemów opartych o banki danych eksploatowane w trybie czasu rzeczywistego.

DRUKOWANIE BEZUDERZENIOWE

W roku 1973 w dalszym ciągu trwał rozwój kopiarek *(hard-copy units)* jako urządzeń wyjściowych, a ich cena spadła do poziomu akceptowanego przez większość użytkowników komputerów. Ostatnio pojawiło się kilka typów drukarek bezdotykowych, między innymi XEROX 1200, który używając elektrofotograficznej techniki obrazowej drukuje 132-znakowe wiersze z prędkością 40 000 wierszy na minutę. Inne drukari tego typu wykorzystują wytryskujące pod ciśnieniem strumienie tuszu, oddziaływanie elektrostatyczne zakończeń przewodów lub wyprowadzenie termiczne poprzez grubą warstwę emulsji.

Generalną tendencją w urządzeniach wyjściowych jest zastosowanie jednego lub wielu biegających promieni (świetlnych lub elektronowych) w celu wygenerowania wzorców składających się z kropek i — w ten sposób — utworzenia znaków alfabetu i innych symboli graficznych.

Tłum. i oprac. **Jan Kiełbasiński** i **Jacek Sobczyk**

---

dok. ze str. 21

P^{6000}_{370} — program P^{PASCAL}_{370} po translacji na CDC 6200

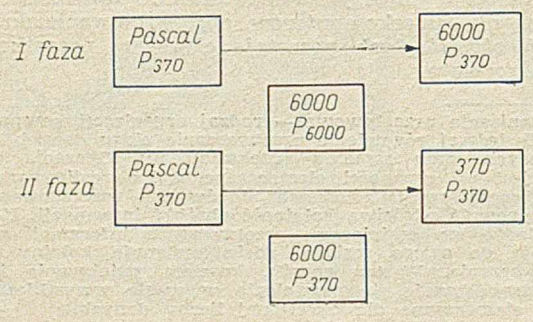
P^{370}_{370} — translator PASCAL-a na kod IBM 370 zapisany w tym kodzie.

Cały proces otrzymania translatora można podzielić na dwie fazy:

gdzie schemat A→C oznacza umieszczenie A jako danych dla programu B i otrzymanie wyniku C.

Jedyną składową kompilatora, którą trzeba zaprogramować 'ręcznie' jest P^{PASCAL}_{370}. Można przy tym wykorzystać pewne fragmenty programu P^{PASCAL}_{6000}.

Rzecz jasna dla otrzymania pracującego kompilatora niezbędne jest napisanie w Assemblerze 370 pakietu programów zapewniających współpracę z systemem OS/370. Są to programy o dużym stopniu trudności, wymagające głębokiej znajomości budowy i działania systemu OS.



BIBLIOGRAFIA

[1] Wirth N.: Programming language PASCAL. Acta Informatica 1971, nr 1.
[2] PASCAL Newsletter: 1974, maj, nr 2.
[3] Dijkstra E. W.: Notes on structural programming. TU Eindhoven, 1969.
[4] Wirth N.: Program development by step-wise refinement , CACM, 1971, rd 12, nr 4.
[5] Wirth N.: Planned changes to the programming language PASCAL. ETH Zurich, 1972 czerwiec.
[6]) Amman U.: Notes accompanying the PASCAL tape, ETH Zurich, 1972.
[7] Wirth N.: The desing of a PASCAL compiler. Software — Practice and experience, 1971, nr 1.
[8] Hoare C. A .R.: An axiomatic definition of the programming language PASCAL. Prec. Symp. on Theoretical Programming, Nowosybirsk, 1972.
[9] Welsh, Quinn: PASCAL compiler for ICL 1900 series computers. Software — Practice and experience, 1972, nr 2

System obrotu lekami

Gdański Zarząd Aptek, jako pierwszy i dotąd jedyny w Polsce powierzył przed pięciu laty obrót hurtowy lekami — gdyńskiemu ZETO. I tak w Dziale Studiów i Projektowania ZETO-Gdynia powstał SAROTL (System automatycznego rozliczania obrotu towarowego lekami). Opracowano go z uwzględnieniem możliwości przetwarzania danych na komputerze ICL-1904. Z uwagi na kompatybilność maszyn system może być również eksploatowany bez jakichkolwiek zmian na komputerach ODRA serii 1300.

Eksploatowany w Gdańskim Zarządzie Aptek od stycznia 1970 roku — system ten ma charakter systemu bazowego, gdyż obejmuje swoim zakresem tematycznym odpowiednio dobrane dziedziny podstawowej działalności przedsiębiorstwa, czyli obrót towarowy w GZA.

SAROTL ma budowę modularną, wobec czego może być rozwijany wg potrzeb użytkownika o nowe moduły lub o nowe zagadnienia przetwarzania informacji w ramach danego modułu.

Obliczenia systemu realizowane są w oparciu o własną bibliotekę programów napisanych w językach PLAN i COBOL (łącznie 56 programów). System wyposażony jest w pełną dokumentację projektową, programową i eksploatacyjną.

W 1973 roku w ZETO — Gdynia system zmodyfikowano tak, że może on być adaptowany przez inne jednostki (przedsiębiorstwa obrotu towarowego) po uprzednim założeniu określonej bazy normatywnej dla danego użytkownika, bez konieczności wprowadzenia dodatkowych zmian w programach. Wdrażanie systemu i przekazywanie do eksploatacji u nowego użytkownika powinno się odbywać etapami, w kolejności poszczególnych jednostek przetwarzania i w miarę organizacyjnego przygotowania przedsiębiorstwa do stosowania systemów informatycznych. W zależności od możliwości organizacyjno-technicznych użytkownika zakłada się kontynuowanie dalszych prac nad rozbudową i doskonaleniem systemu w zakresie:

— optymalizacji środków transportu i powierzchni składowania towarów
— automatycznego wydawnictwa indeksu i katalogów towarowych
— wprowadzenia transmisji danych i urządzeń ekranowych do „konwersacji" oraz bieżącej aktualizacji zbiorów informacji (indeksów)
— wprowadzenie dalszych udoskonaleń wynikających z doświadczeń eksploatacyjnych oraz nowych, rosnących potrzeb użytkownika systemu.

Charakterystyka ogólna użytkownika systemu

Gdański Zarząd Aptek jest przedsiębiorstwem wielozakładowym, działającym na zasadzie pełnego rozrachunku gospodarczego. W ramach swojej działalności podstawowej prowadzi on:

— 118 aptek sprzedaży detalicznej (rocznie otwiera trzy nowe apteki)

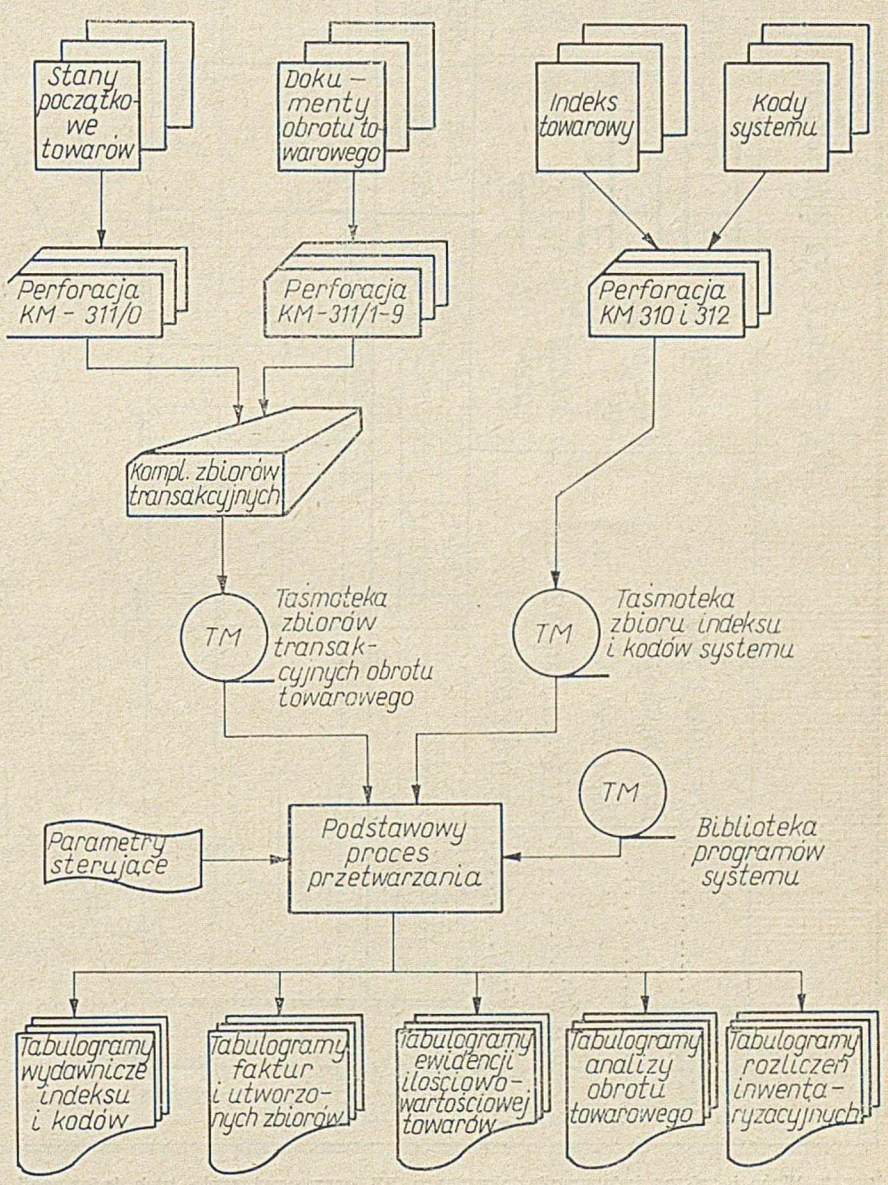
— 95 punktów aptecznych w szpitalach, sanatoriach, klinikach i instytutach naukowo-badawczych

— hurtowy zakup towarów od ok. 220 dostawców i producentów leków
— magazynowanie (przechowywanie) towarów — leków w 18 magazynach branżowych hurtowni

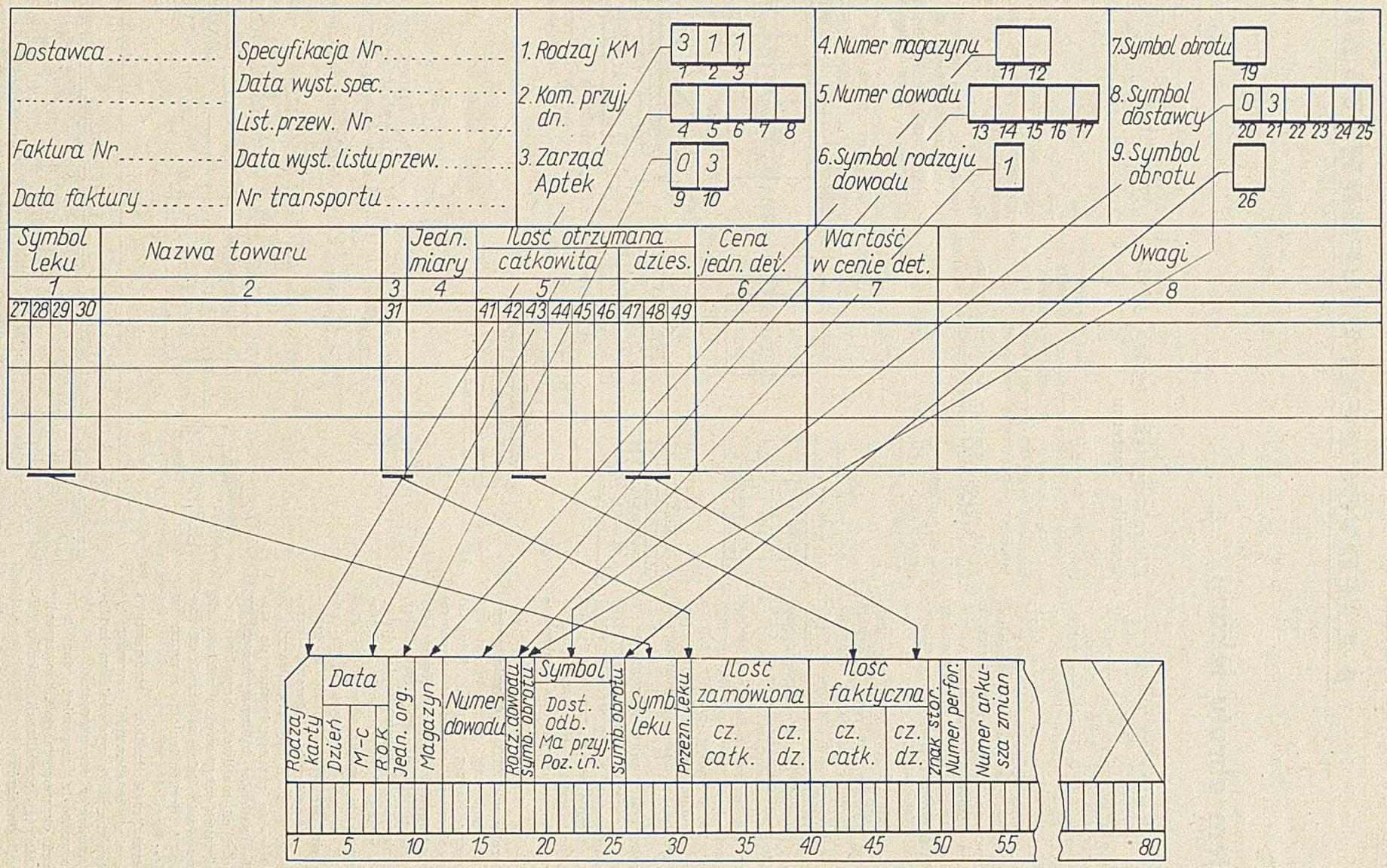
— sprzedaż hurtową leków dla ok. 360 stałych odbiorców

— zaopatrzenie własnych aptek zlokalizowanych na terenie całego województwa

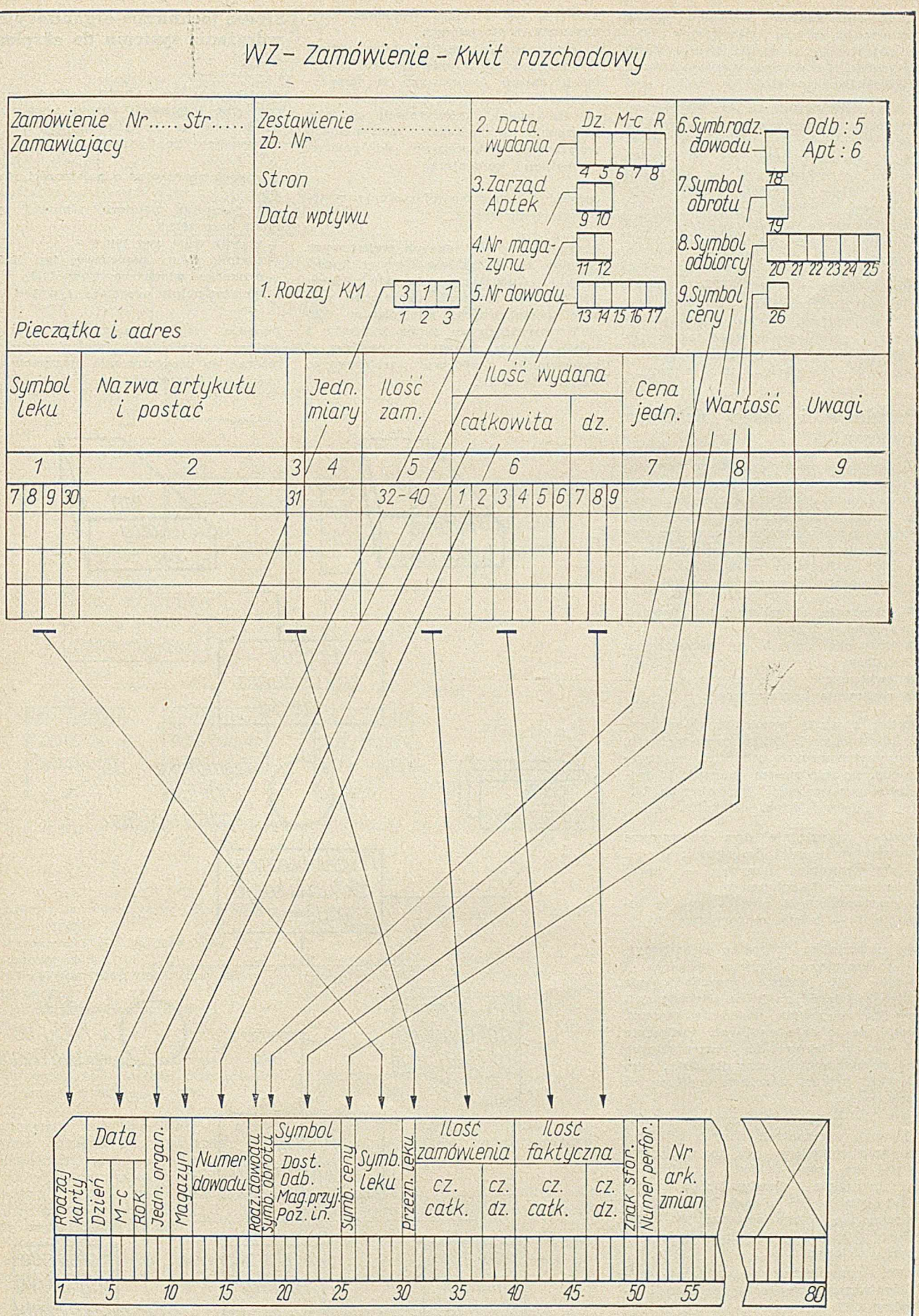
— produkcję oraz fasowanie leków we własnym laboratorium galenowym
— zaopatrzenie w materiały pomocnicze zarządu, laboratorium galenowego,



Rys. 1. Schemat ideowy funkcjonowania systemu SAROTL



Rys. 2. Wzór i zakres informacji dokumentu PZ oraz powiązanie tych informacji z kartą dziurkowaną



Rys. 3. Wzór i zakres informacji dokumentu WZ oraz ich powiązania z kartą dziurkowaną

aptek oraz innych jednostek służby zdrowia na terenie województwa
— zaopatrzenie w środki farmaceutyczne odbiorców innych województw w ramach przerzutów.

Tak szeroka działalność przedsiębiorstwa wymaga rozliczania masy towarowej w różnych cenach (hurtu, detalu, przerzutu, zbytu) w zależności od szczebla obrotu towarowego lub też produkcji leków. Ceny, a zatem i ich marże handlowe, wymagają wielu operacji arytmetycznych komputera oraz opracowania licznych tabulogramów wynikowych.

Indeks towarowy leków (prowadzony przy użyciu komputera) obejmuje około 6.000 pozycji. Liczba podstawowych operacji obrotu towarowego leków (zakupu i sprzedaży) wynosi miesięcznie ponad 70 tys. (około 5 mln znaków).

Przedmiot i zakres działania systemu

Przedmiotem systemu SAROTL jest automatyzacja ewidencji, rozliczeń, fakturowania i analizy obrotu towarowego lekami w Gdańskim Zarządzie Aptek. System został podzielony na cztery zintegrowane podsystemy — moduły, obejmujące podstawowe zagadnienia z zakresu obrotu towarowego lekami:

- zakładanie, aktualizację i wydawnictwo indeksu leków
- kontrolę danych i fakturowanie sprzedaży
- ewidencję
- rozliczenia inwentaryzacji.

Każdy z wyżej wymienionych podsystemów obejmuje kilka samodzielnych, zautomatyzowanych jednostek przetwarzania. Łącznie system obejmuje 15 jednostek przetwarzania (omówionych bliżej w dalszej części artykułu).

System uwzględnia dwa podstawowe maszynowe nośniki informacji:
— przygotowania danych — karty dziurkowane 80-kolumnowe
— przetwarzania i przechowywania informacji — taśmy magnetyczne.

Za najmniejszą jednostkę organizacyjno-rozliczeniową przyjęto magazyn i aptekę (tj. miejsce przechowywania towarów-leków), za które odpowiedzialny jest kierownik magazynu (apteki). Założenie to daje możliwość rozliczenia i analizy leków dla poszczególnych miejsc składowania oraz dla GZA, jako całości. System rozróżnia następujące rozliczenia:

„C" — cykliczne (praktycznie — codzienne), do kontroli danych i fakturowania sprzedaży. (Są to rozliczenia dwu- lub trzy- dniowe w zależności od aktualnie obowiązującego cyklu fakturowania. Po okresie wdrażania i przekazania całego systemu do eksploatacji stosuje się fakturowanie bieżące codzienne)

„M" — miesięczne, do ewidencji ilościowo-wartościowej oraz rozliczeń i analizy OTL

„O" — okresowe, do opracowywania spisów z natury oraz rozliczania różnic inwentaryzacyjnych (inwentaryzacje planowe i pozaplanowe).

SAROTL jest również systemem ZINTEGROWANYM poprzez:
— wykorzystanie wspólnej bazy danych
— wzajemne powiązanie zbiorów informacji obrotu towarowego w poszczególnych podsystemach
— wielokrotne wykorzystanie zbiorów informacji w różnych podsystemach wczytanych jednorazowo do pamięci zewnętrznej komputera.

Ideowy schemat funkcjonowania systemu ilustruje rys. 1.

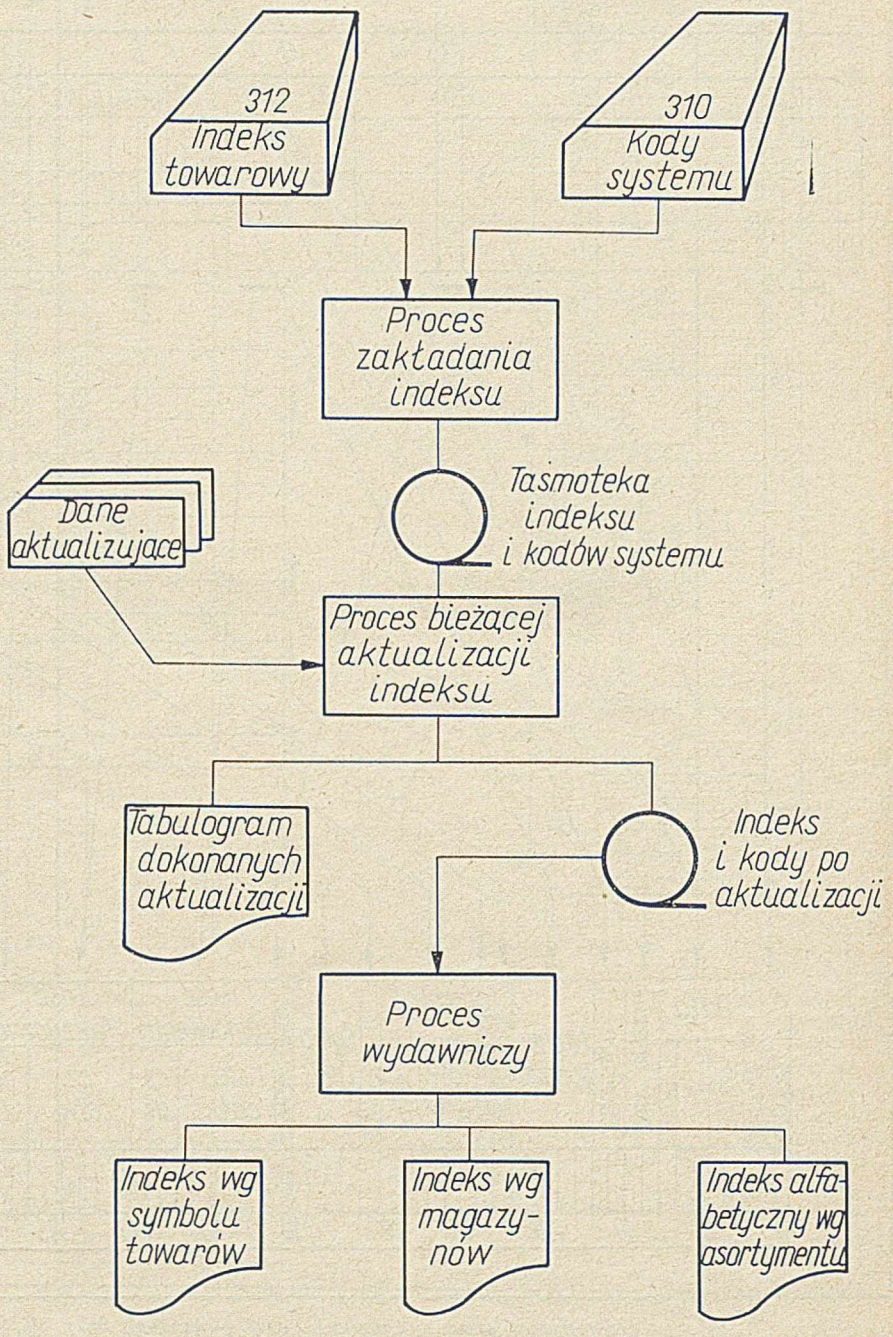
Przygotowywaniem danych wejściowych zajmuje się Zakładowy Ośrodek Informatyki Gdańskiego Zarządu Aptek. Ośrodek ten zatrudnia operatorów maszyn pomocniczych, kontrolerów WE/WY, inspektora do spraw indeksów i kodów (mgra farmacji), analityka i pracowników nadzoru technicznego (łącznie 14 osób).

Środki techniczno-organizacyjne wdrażania systemu do eksploatacji

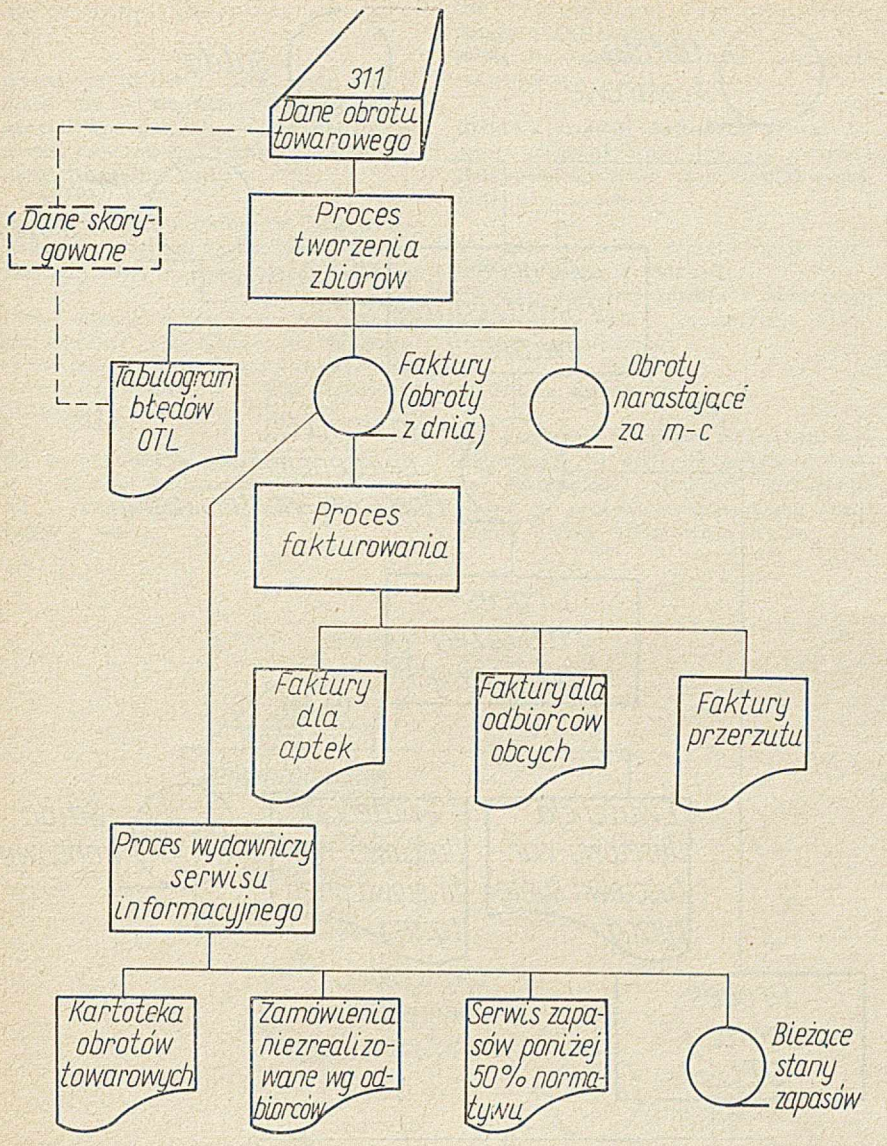
Projekt systemu SAROTL zakłada przetwarzanie informacji obrotu towarowego lekami na komputerze ICL-1904 o następującej konfiguracji:

— pamięć operacyjna o pojemności 16 K słów
— 6 jednostek pamięci taśmowej typ 1973/1 oraz 1973/2
— czytnik kart typ 1912/9
— czytnik taśmy papierowej typ 1916/2
— drukarka wierszowa typu 1933
— monitor-pulpit operatora typ 1904/5.

Ponadto eksploatacja systemu zakłada wykorzystanie niżej wymienionych urządzeń do przygotowania maszynowych nośników informacji:



Rys. 4. Zakładanie i aktualizacja indeksu leków



Rys. 5. Kontrola danych transakcyjnych, fakturowanie oraz serwis informacyjny

— dziurkarka numeryczna kart 80-ciokolumnowych
— sprawdzarka numeryczna kart 80-ciokolumnowych
— dziurkarka alfanumeryczna kart z opisywaczem
— dziurkarka taśmy papierowej 8-ścieżkowej.

Jak już wspomniano, eksploatacja systemu może się również odbywać na komputerze ODRA serii 1300.

Użytkownik — po zapoznaniu się z projektem ogólnym i technicznym oraz dokumentacją programową i eksploatacyjną systemu SAROTL — powinien przystąpić do wykonania prac organizacyjnych w przedsiębiorstwie, z których najistotniejszymi są:

❶ dostosowanie indeksu towarowego do wymogów systemu EPD

❷ opracowanie symboliki kodowania informacji w systemie (klucz dostawców, odbiorców, komórek organizacyjnych, cen, postaci leków, grup farmakologicznych, jednostek miary itp.)

❸ wdrożenie nowych dokumentów obrotu towarowego lekami

❹ wprowadzenie niezbędnych zmian organizacyjnych wynikających z przejęcia przez Zakładowy Ośrodek EPD wielu czynności związanych z przetwarzaniem informacji, dotychczas przetwarzanych w innych komórkach

❺ zorganizowanie Zakładowego Ośrodka Przetwarzania Danych, adaptacja lokalu, uruchomienie maszyn perforujących i przeszkolenie operatorek obsługi technicznej na specjalnie zorganizowanym kursie

❻ przygotowanie przeszkolenia szeregu pracowników poszczególnych komórek organizacyjnych do korzystania z tabulogramów komputerowych jako nowych nośników informacji służących do podejmowania decyzji

❼ przygotowanie zainteresowanych pracowników przedsiębiorstwa do możliwości dysponowania zbiorami informacji w pamięci zewnętrznej komputera w ramach poszczególnych podsystemów

❽ przestrzeganie dyscypliny przygotowania i przetwarzania danych w przyjętych do realizacji instrukcjach technologiczno-eksploatacyjnych, stanowiących przedmiot odrębnej dokumentacji systemu.

Zasadniczym warunkiem wdrożenia systemu do eksploatacji jest etapowość wdrażania poszczególnych jednostek przetwarzania dla danego podsystemu. Podjęcie prac nowego etapu — nowej jednostki przetwarzania — wymaga pomyślnego zakończenia prac poprzedniego etapu — poprzedniej jednostki przetwarzania. Zasada ta była w pełni realizowana zarówno przez zinteresowanych pracowników GZA, jak i cały zespół projektowo-programowy systemu.

Przygotowanie danych wejściowych

❶ Dokumenty obrotu towarowego

Po przeprowadzeniu unifikacji i modyfikacji dokumentów obrotu towarowego udział w przetwarzaniu biorą obecnie informacje z następujących dokumentów źródłowych:

— stany początkowe — inwentaryzacja — symbol dokumentu 0

— PZ kwit przychodowy — symbol dokumentu 1

— PZ zwroty od odbiorców — symbol dokumentu 2

— PN protokół nadwyżek (inne przych.) — symbol dokumentu 3

— PK strona dodatnie — symbol dokumentu 4

— WZ kwit rozchodowy (zamówienia odbiorców) — symbol dokumentu 5

— WZ kwit rozchodowych (zamówienia aptek) — symbol dokumentu 6

— PU protokół ubytków (inne rozchody) — symobl dokumentu 7

— PK strona ujemne — symbol dokumentu 8

— Mn przesunięcia towarów między magazynami — symbol dokumentu 9

Podaje się przykładowo wzory dwóch podstawowych dokumentów obrotu towarowego (rys. 2 i 3), zakres informacji w tych dokumentach oraz ich powiązanie z maszynowymi nośnikami informacji (karty dziurkowane).

❷ Indeks towarowy (leków)

Indeks leków został opracowany w układzie alfabetycznym wg asortymentu leków, z odpowiednią rezerwą symboli leków dla każdej litery alfabetu. Poszczególne informacje indeksu leków oznaczone są symbolami cyfrowymi z wyjątkiem nazwy leków, gdzie występują znaki alfabetyczne (23 znaki).

Korekty i uzupełnienia indeksu leków, z uwagi na zachowanie jednolitej nomenklatury, mogą być dokonywane wy-

łącznie za zgodą Zakładowego Ośrodka Informatyki. Treść indeksu leków obejmuje następujące informacje, aktualizowane bieżąco (codziennie), zgodnie z harmonogramem fakturowania i rozliczania obrotu towarowego:

1 — symbol leku: 4 znaki numeryczne, jako stały identyfikator danych asortymentu indeksu leków

2 — podstawa prawna: 1 znak numeryczny, zgodnie z kluczem nr 13

3 — grupa farmakologiczna: 2 znaki numeryczne, zgodnie z kluczem nr 13

4 — postać leku: 2 znaki numeryczne, zgodnie z kluczem nr 14

5 — określenie nazwy leku: 23 znaki alfanumeryczne wg indeksu leków

6 — jednostka miary: 1 znak numeryczny, zgodnie z kluczem nr 15

7 — cena detaliczna: 9 znaków numerycznych, w tym 6 na złote i 2 na grosze

8 — marża detaliczna: 4 znaki numeryczne określające wysokość marży wyrażonej w procentach (2 miejsca całkowite i 2 miejsca dziesiętne)

9 — marża hurtowa: 4 znaki numeryczne określające wysokość marży wyrażonej w procentach (2 miejsca całkowite i 2 miejsca dziesiętne)

10 — normatyw interwencyjny zapasów: 9 znaków numerycznych na określenie ilościowego zapasu normatywnego, w tym 6 znaków na ilość całkowitą i 3 znaki na ilość dziesiętną

11 — towary specjalne: 1 znak alfanumeryczny

12 — miejsce składowania: 2 znaki numeryczne

13 — inne dane wg potrzeb użytkownika

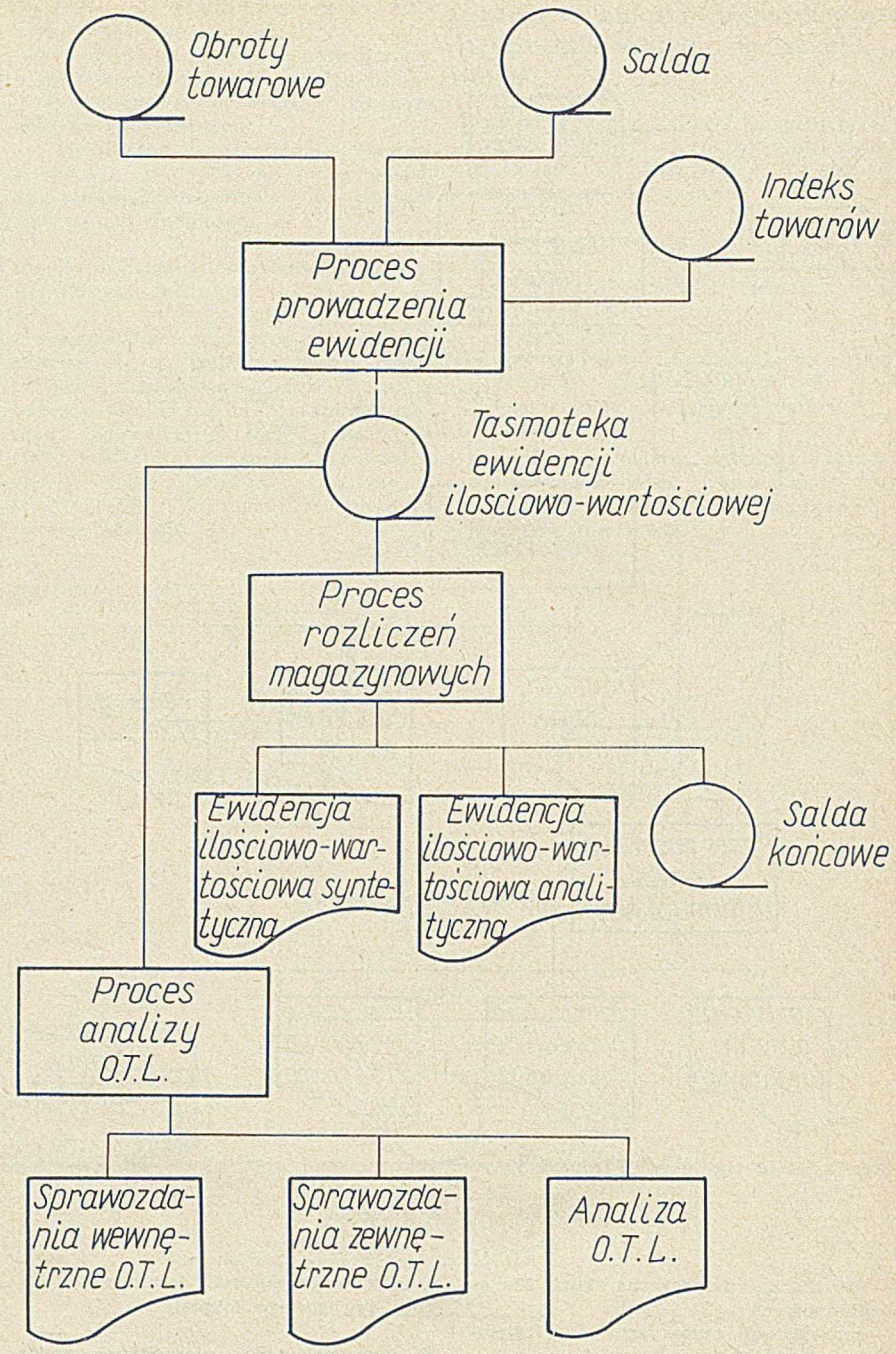
● Kody systemu

W systemie SAROTL do symbolizacji danych na dokumencie źródłowym, karcie maszynowej i w procesie przetwarzania stosuje się następujące kody-klucze:

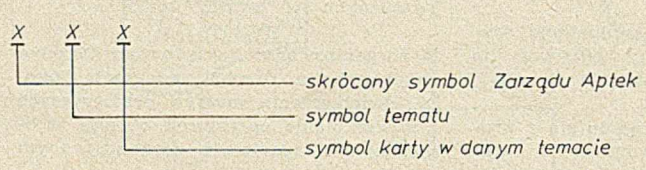
klucz kart dziurkowanych 1
klucz wojewódzkich zarządów aptek 2
klucz magazynów — komórek organizacyjnych 3
klucz dokumentów źródłowych 4
klucz obrotu towarowego 5
klucz dostawców 6
klucz odbiorców 7
klucz cen 8
klucz przeznaczenia leku 9
klucz storna 10
klucz rodzaju leków — indeks 11
klucz podstawy prawnej leku 12
klucz grupy farmakologicznej 13
klucz postaci leku 14
klucz jednostek miary 15

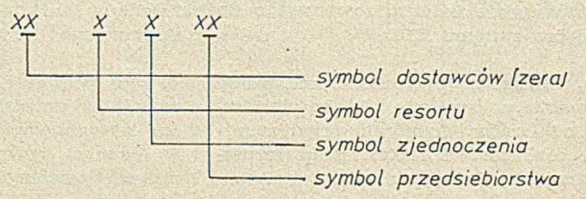
Przykładowo podaje się strukturę klucza 1 i 6.

Klucz kart dziurkowanych (klucz 1). Klucz rodzaju kart maszynowych jest



Rys. 6. Prowadzenie ewidencji ilościowo-wartościowej i analiza obrotów towarowych





trzymiejscowy i ma następującą budowę:

Występują tu następujące karty maszynowe: 310 — dla kluczy do kodowania danych systemu, 311 — dla danych obrotu towarowego lekami, 312 — dla danych indeksu leków.

Klucz dostawców (klucz 6). Klucz dostawów opracowany jest jako symbol sześciocyfrowy o następującej budowie:

Układ klucza dostawców pozwala na zakodowanie współzależności organizacyjnej pomiędzy jednostką będącą bezpośrednim dostawcą z jego jednostką nadrzędną i resortem.

Np. symbol 006104 oznacza:
00 ... Symbol dostawców
006 ... Ministerstwo Przemysłu Spożywczego i Skupu
0061 ... Zjednoczenie Przemysłu Zielarskiego HERBAPOL
006104 Poznańskie Zakłady Zielarskie HERBAPOL

Niektóre symbole dostawców mogą występować dwukrotnie, jak np. Centrala Handlowa Sprzętem Medycznym występuje w grupach

001 jako dostawca sprzętu medycznego sprzedawanego w aptekach
009 jako dostawca sprzętu stanowiącego wyposażenie aptek (zaopatrzenia materiałowo-technicznego).

● Zbiory informacji

Ze wszystkich zbiorów informacji występujących w systemie wyróżnić można następujące taśmoteki, szczególnie ważne w procesie przetwarzania rozliczania obrotu towarowego lekami:

1) IB119 INDEKS — taśmoteka stała, obejmuje informacje indeksu leków i kluczy do kodowania danych. Podlega bieżącej aktualizacji w oparciu o „Raporty zmian"

2) IB119 FAKTUR — taśmoteka informacji do procesu fakturowania za dany cykl rozliczeniowy (1, 2 lub 3 dniowy)

3) IB119 OBROTY — taśmoteka codziennych operacji obrotu towarowego lekami

4) IB119 SALDA — taśmoteka sald początkowych z rozliczenia operacji z poprzedniego miesiąca

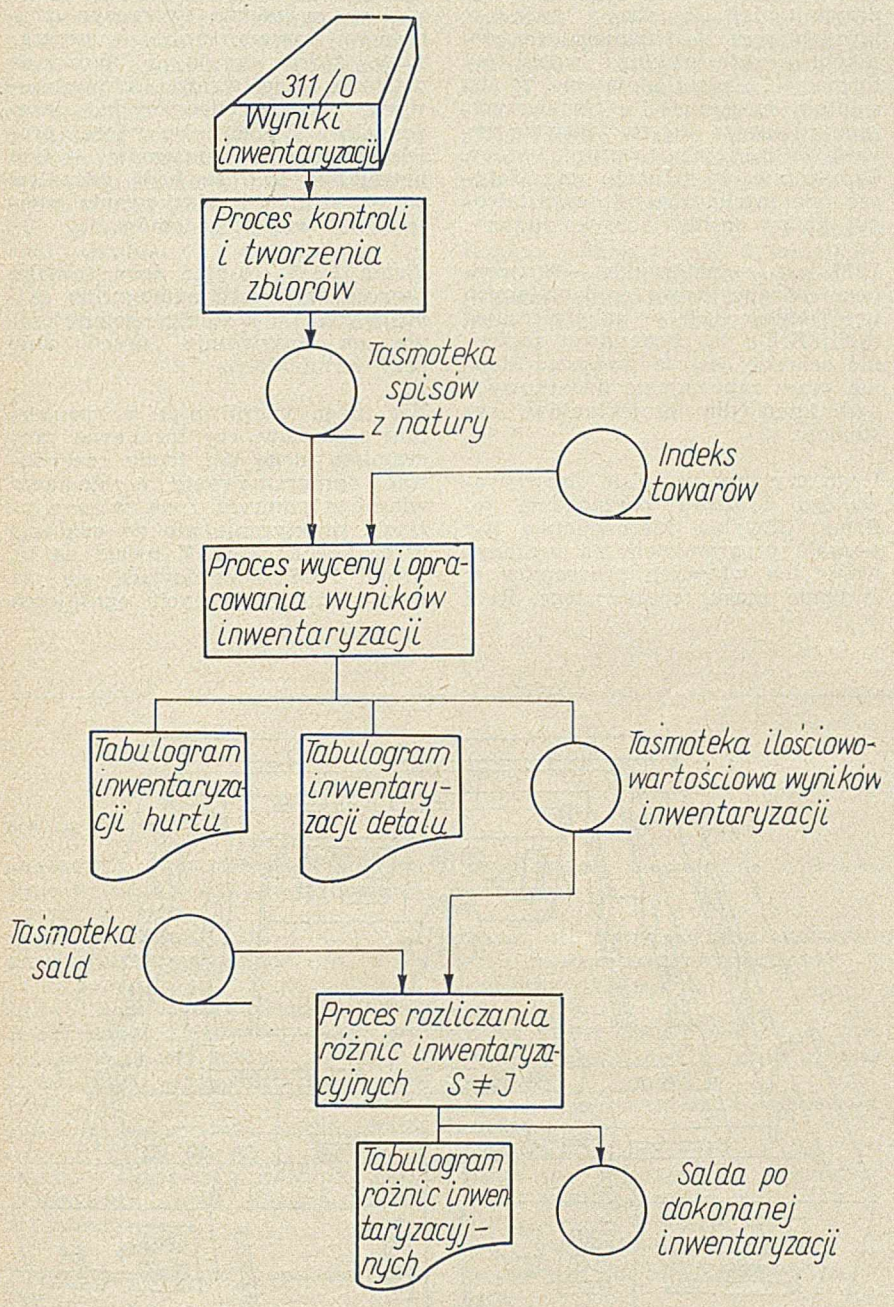
5) IB119 ZAPASY — taśmoteka ewidencji obrotu towarowego za dany dzień

6) IB119 OTL WA — taśmoteka ewidencji ilościowo-wartościowej OTL za dany miesiąc sprawozdawczy

7) IB119 INWENT — taśmoteka na dopisywanie informacji spisów z natury

8) IB119 INW TE — taśmoteka ilościowo-wartościowa inwentaryzacji

9) IB119 INW TH — taśmoteka stanów inwentaryzacyjnych, która po zmianie na MT „IB119 SALDA" stanowi stan początkowy do ewidencji towarów.



Rys. 7. Rozliczenia wyników inwentaryzacji

Zestawienia wynikowe systemu

W wyniku realizacji określonych programów otrzymuje się podstawowe zestawienia wynikowe (tabulogramy) systemu (wymienimy tylko niektóre z nich):

W zakresie indeksu leków: tabulogram wydruku kluczy systemu i indeksu leków w układzie: symboli towarów, magazynów i nazwy asortymentu — wydawnictwo całkowite (lekospis), w nakładzie wewnętrznym (również dla klienta) oraz tabulogram dokonanych korekt (bieżącej aktualizacji) kluczy i indeksu leków.

W zakresie kontroli i fakturowania: faktury dla odbiorców własnych i obcych i przerzuty towarowe oraz serwis informacyjny towarów poniżej 50% normatywu oraz zamówień niezrealizowanych wg odbiorców.

W zakresie ewidencji i analizy OTL: ewidencja ilościowo-wartościowa obrotów i stanów towarowych oraz tabulogramy analizy obrotu towarowego lekami dotyczące realizacji zamówień, wykonania normatywów, analizy grup marżowych, grup farmakologicznych i grup odbiorców.

W zakresie rozliczania wyników inwentaryzacji: tabulogram ilościowo-wartościowy spisów z natury magazynów i aptek oraz tabulogram różnic inwentaryzacyjnych wg miejsc składowania i rodzaju leków.

Organizacja procesu technologicznego przetwarzania danych

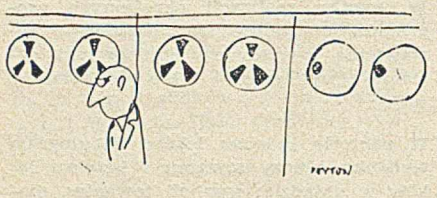
Proces technologiczny przetwarzania systemu SAROTL został podzielony na

c.d. na str. 33

Problemy socjologiczne systemów komputerowych

Przedsięwzięcia mające na celu ochronę komputerów zmieniły radykalnie stosunki i warunki pracy personelu instytucji zajmujących się przetwarzaniem danych. Jak na te zmiany zareagują pracownicy elektronicznego przetwarzania danych? Oto pytanie, które powstaje w związku z koniecznością zabezpieczenia programów i systemów przed destrukcyjnymi lub niezgodnymi z przeznaczeniem tych systemów działaniami, nie tylko osób spoza ośrodka, ale także i jego personelu.

Odpowiedzi na te i inne problemy związane z niebezpieczeństwami grożącymi bezpośrednio ludziom zatrudnionym w ośrodkach komputerowych i samym systemom komputerowym — próbują szukać dr Edward Harriman (ostatnio ukończył raport poświęcony problemom banków danych dla Krajowej Rady do spraw Swobód Obywatelskich) i Hedley Voysey w swoim artykule w brytyjskim czasopiśmie NEW SCIENTIST (nr 896 z 2 maja 1974 r.).



O randze problemu świadczy zainteresowanie okazywane przez przemysł urządzeń i systemów zabezpieczających przez liczne firmy komputerowe (w tym i tak potężne, jak IBM), a nawet — Narodowe Centrum Komputerowe. Wydaje się nie ulegać wątpliwości, iż dodatkowa ochrona ośrodka jest koniecznością, choćby nawet niektóre pociągnięcia związane z jej wprowadzeniem miały naruszyć dobre stosunki między pracownikami i kierownictwem. Takim posunięciem może być świadome, bardzo szczegółowe rozbicie zadań, wprowadzenie specjalnie przygotowanych pomieszczeń komputerowych, ciągła kontrola i nadzór nad personelem, co razem czy z osobna wzięte nie jest pozbawione tych problemów socjo- i psychologicznych, których mamy prawo poważnie się obawiać: pogarszają się stosunki załoga-kierownictwo, pojawia się zjawisko zwane dehumanizacją miejsca pracy.

Powszechnie ostatnio stosowanym rozwiązaniem, jeśli chodzi o fizyczną stronę realizacji zadań ochrony komputera, jest tzw. sala maszyny — bez okien i z jednym dobrze strzeżonym wejściem. Skomplikowane kontrole dokumentów przy wejściu mają nie dopuścić do pokoju osób niepowołanych. Samo zaś pomieszczenie jest podzielone na małe pokoiki, w których personel przetwarzania danych wykonuje swoje szczegółowe zadania, wymieniając między sobą materiały (takie, jak np. dyski) i dokumenty przez specjalne otwory w ścianach i porozumiewając się tylko za pomocą prostych, lecz zakodowanych sygnałów. **Zaleca się dalej, aby załoga nie utrzymywała bliskich kontaktów towarzyskich między sobą,** a także aby pracę rozdzielać pomiędzy mieszkańców różnych geograficznie obszarów.

Oczywiście, wszystkie te poczynania mają zapewnić bezpieczeństwo, przynajmniej ze strony tych w których ręce — zdaniem kierowników — nie powinna wpaść informacja. Ale też stoją one w absolutnej sprzeczności z tym wszystkim, co uważa się za właściwe zawodowe i socjalne warunki pracy. Szczególnie w sytuacji, gdy dotyczą one inteligentnej i wykształconej kadry obsługującej komputery.

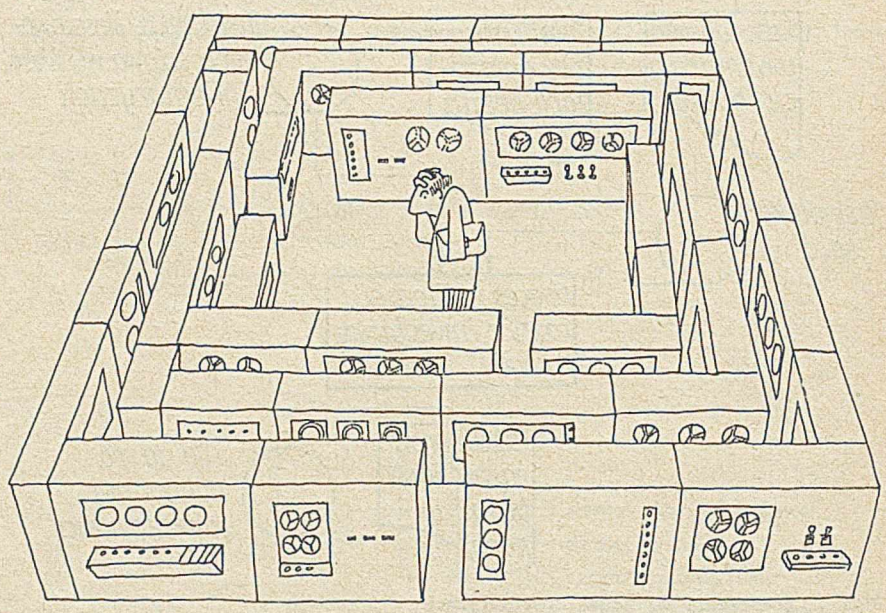
IBM jest — jak zwykle — koncernem wiodącym we wspomnianych przedsięwzięciach zabezpieczających; firma ta chce nawet zapewnić ochronę jeszcze wyższego stopnia przez zapewnienie dodatkowych ograniczeń dla projektantów systemów.

I oto czytamy w opisie Zaawansowanego Systemu Zarządzania tej firmy: „Zwykle, kierownictwo nie zezwala programistom na wykonywanie ich własnych programów w systemie czasu rzeczywistego. Robi się tak po to, aby uniknąć ryzyka związanego z wykonywaniem opcji nieopisywanych, różnych od założonych dla systemu. Możliwe byłoby wtedy takie sterowanie programem lub grupą programów, aby wykonywały one coś więcej ponad ich planowanymi zakresami. I tak np. wprowadza się strumienie informacyjne dla systemów w taki sposób, żeby twórca programu nadrzędnego sterującego systemem nie mógł używać systemu czasu rzeczywistego bez nadzoru kierownictwa".

Następstwa tego podejścia muszą się wydawać trudne do zniesienia dla rzetelnych i zaangażowanych w swoją pracę takich programistów, którzy zamierzają poświęcić swoje kariery zawodowe tworzeniu programów systemowych — programów, które sterują i nadzorują działanie zestawów komputerowych. I tak dochodzimy do paradoksalnej sytuacji: **im więcej projektanci stworzą systemów — tym pewniejsze jest, że będą odsunięci od wdrażania i użytkowania ostatecznych wersji systemów.**

Firma IBM stosując taką taktykę skazuje większość naukowców zajmujących się techniką komputerową na dożywotnie urzędowanie przy terminalach.

Nie ulega wątpliwości, że projektant systemów, czy też nawet programista mogą bez trudu rozbudować oprogramowanie o elementy własnego pomysłu i na własny użytek, praktycznie nie do odkrycia przez kogokolwiek. Z uwagi na to, firma IBM zdecydowała się na wprowadzenie licznych ograniczeń



dla personelu jako drogę rzekomo prowadzącą do właściwego budowania systemów.

Zdziwienie może wywołać fakt wyjątkowo słabej, można rzec — nieśmiałej reakcji społeczności specjalistów komputerowych na posunięcia ich firmy, czy też towarzystw zajmujących się techniką i filozofią zabezpieczeń. Wydaje się, że z jednej strony chcieliby oni zachować swoje stanowiska pracy, a z drugiej — zupełnie słusznie — walczą o zachowanie poczucia godności osobistej. W społeczności tej panuje zgodna opinia co do tego, że **należy domagać się ustalenia własnego statusu zawodowego** (co prawdopodobnie pociągnie za sobą wysuwanie roszczeń wykazujących przypadkową zbieżność z interesami kierownictwa). Nie chce ona być traktowana na równi z kadrą urzędniczą i administracyjną, którą zastępuje. Ale jest jedna przeszkoda w tych dążeniach — uważa się, że personel przetwarzania danych nadużył zaufania swego kierownictwa. A ponieważ jednocześnie przemysł urządzeń zabezpieczających kładzie coraz większy nacisk na sprawę komputerów — pozycja zawodowa techników komputerowych będzie niewątpliwie ograniczana. Coraz powszechniejsza staje się opinia, że technicy powinni ulec, posłusznie zezwalając na wprowadzenie ograniczeń, jak również to, że wcześniej nie powinni byli nadużywać swych możliwości zawodowych. Nieuchronnym skutkiem nowej recesji na komputerowym rynku pracy było by nasilenie się postaw konformistycznych. Personel będzie się bowiem podporządkowywał restrykcjom, chcąc zachować pracę.

Jest jeszcze jeden problem ważny, który powinniśmy rozważyć — problem odpowiedzialności pracowników firm komputerowych. Czy odpowiedzialność ta aktualnie rośnie, czy też maleje? Jaka jest możliwa odpowiedź personelu przetwarzania danych?

Można tu podać dobry przykład, ilustrujący tezę zwiększonej odpowiedzialności — demokratyczny sposób wykorzystywania komputerów wymaga nie tylko publicznej odpowiedzialności kierownictwa, ale także istotnie zdecentralizowanej, nie hierarchicznej struktury podejmowania decyzji. Znaczy to, że osoby obsługujące komputery są częściowo odpowiedzialne i akceptują następstwa tych sposobów użytkowania, nad którymi pracują. Ma to posmak nawoływania do ruchu związków zawodowych lub może nawet udziału pracowników w zarządzaniu w przemysłach komputerowym oraz urządzeń i systemów zabezpieczających.

Trudno jednak mówić o jakiejkolwiek decentralizacji zarządzania w firmach komputerowych; trudno też o podobne postawy w przemyśle zabezpieczeń, gdzie zatrudnia się byłych policjantów i funkcjonariuszy wojskowej służby bezpieczeństwa, którzy teraz zajmować się będą stosunkami pracy. Także Brytyjskie Towarzystwo Komputerowe wydaje się wykazywać brak zainteresowania. Wydany przez nie „Kodeks Pracy" wprost nakazuje personelowi działać wyłącznie zgodnie z celami kierownictwa i w rezultacie przekształca pracowników w służbę bezpieczeństwa ich pracodawców. W takiej sytuacji nie można chyba wymagać owej zwiększonej odpowiedzialności pracowników za sposób wykorzystania komputerów.

Przez długi czas dążenie do ochrony komputerów było reklamowane publicznie jako wspaniałomyślny wkład przemysłu komputerowego w szeroko rozumianą ochronę sfery osobistej człowieka. Ale społeczeństwo nie dało się zwieść tym frazesom i dość szybko odkryło, że tajemnice i sekrety rządu i korporacji oraz prywatne interesy przemysłu komputerów są całkowicie zbieżne.

W interesie społecznym leży więc odzyskanie kontroli nad tą dziedziną życia społecznego. Ciekawe, jak długo jeszcze personel przetwarzania danych będzie dawał się zwodzić w podobny do dotychczasowego sposób?

Tłumaczył i opracował

Michał Bojańczyk

---

dok. ze str. 31

następujące samodzielne, zautomatyzowane jednostki przetwarzania (moduły):

Moduł I: ZAKŁADANIE I AKTUALIZACJA INDEKSU LEKÓW (rys. 4)

JP-1.1. Proces zakładania indeksu leków
JP-1.2. Proces przygotowania kart korygujących indeks leków
JP-1.3. Proces korygowania indeksu leków
JP-1.4. Proces wydruku indeksu leków

Moduł II: KONTROLA DANYCH, FAKTUROWANIE ORAZ SERWIS INFORMACYJNY (rys. 5)

JP-2.1. Proces kontroli i tworzenia zbiorów
JP-2.2. Proces fakturowania i serwisu informacyjnego.

Moduł III: PROWADZENIE EWIDENCJI ILOŚCIOWO-WARTOŚCIOWEJ I ANALIZA OTL (rys. 6)

JP-3.1. Proces obliczeń ilościowych OTL
JP-3.2. Proces korekty merytorycznej OTL
JP-3.3. Proces obliczeń wartościowych i analizy OTL

Moduł IV: ROZLICZENIA WYNIKÓW INWENTARYZACJI (rys. 7)

JP-4.1. Proces zakładania taśmoteki na dopisywanie danych inwentaryzacyjnych
JP-4.2. — Proces kontroli i tworzenia zbiorów inwentaryzacji
JP-4.3. Proces opracowania spisów z natury
JP-4.4. Proces korekty merytorycznej danych inwentaryzacji
JP-4.5. Proces opracowania skorygowanych spisów z natury
JP-4.6. Proces obliczania różnic inwentaryzacyjnych.

Taki podział procesu technologicznego pozwala na systematyczne wdrażanie i przekazywanie poszczególnych jednostek systemu do eksploatacji. Ponadto opisana organizacja zapewnia automatyczną kontrolę danych i kontrolę procesu przetwarzania poszczególnych jednostek systemu.

Józef Czubek

JANUSZ GRZEMOWSKI
Instytut Maszyn Matematycznych
Oddział Śląski
Katowice

681.322.06:681.327.2.004.14:651.8:658.5:621

Oprogramowanie Jednolitego Systemu System kartotek bazowych

Omówiono strukturę kartotek i ich powiązania, jak programy użytkowe i zestawy makroinstrukcji. System kartotek bazowych jest przeznaczony dla przedsiębiorstw przemysłowych do celów technicznych przygotowania produkcji seryjnej. Może być wykorzystany pod kontrolą DOS-JS (minimum 48 KB pamięci operacyjnej i 2 jednostki pamięci dyskowej po 7,25 MB).

System kartotek bazowych umożliwia tworzenie bazy danych dla przedsiębiorstw lub innych jednostek gospodarki w zakresie technicznego przygotowania produkcji i stanowi podstawę do wdrażania w naszym przemyśle zautomatyzowanych systemów zarządzania w takich dziedzinach, jak: gospodarka materiałowa, planowanie produkcji i zakupów, rachunek kosztów itp.

Struktura kartotek i wzajemne powiązania zawartych w nich informacji pozwalają w prosty sposób na automatyczne obliczenie:

— zapotrzebowania na surowce, półfabrykaty i części niezbędne do produkcji
— obciążenia stanowisk pracy
— zapotrzebowania na narzędzia
— kosztów i pracochłonności produkcji.

Realizacja wszystkich wymienionych funkcji jest możliwa dzięki odpowiedniej strukturze oprogramowania.

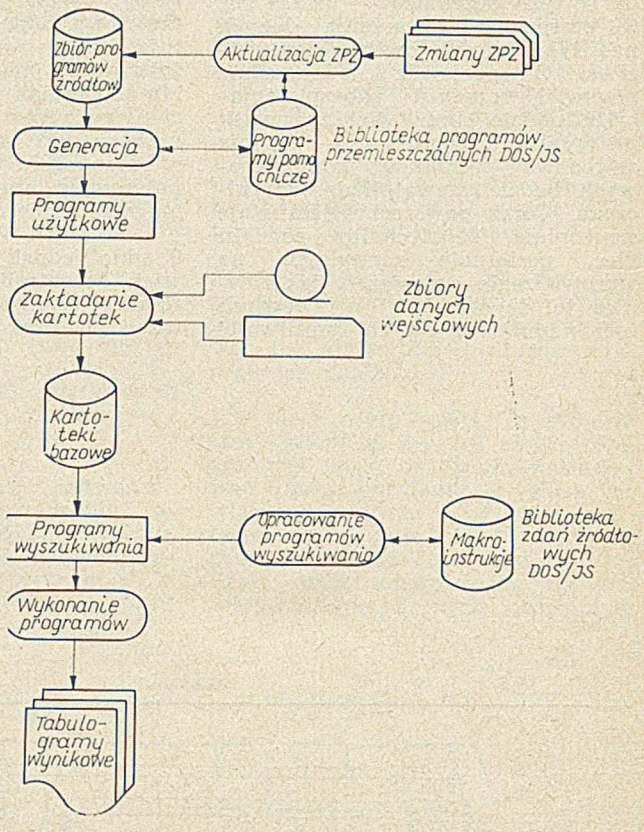
Oprogramowanie systemu

Oprogramowanie składa się z generatora programów użytkowych, które zakładają utrzymują i reorganizują kartoteki użytkownika, oraz z zestawu makroinstrukcji ułatwiających tworzenie programów wyszukiwania i przetwarzania informacji z kartotek bazowych (patrz rys. 1).

Generatorem programów nazywamy zbiór programów pomocniczych (z których tworzony jest program generujący) i zbiór tekstów źródłowych (ZTZ).

Programy pomocnicze tworzone są przez kombinację 10 modułów przechowywanych w bibliotece programów przemieszczalnych DOS/JS. Służą one do:

— wyprowadzania ZTZ z dysku systemowego na taśmę, karty lub drukarkę
— wprowadzenia ZTZ na dysk z kart lub taśmy
— aktualizowania ZTZ, tj. dodawania, usuwania i zmieniania instrukcji
— generowania oprogramowania z tekstu ZTZ.



Rys. 1. Schemat ogólny przetwarzania kartotek bazowych

Zbiór tekstów źródłowych (ZTZ) składa się z procedur napisanych w języku ASSEMBLER i przechowywany jest na dysku. Niektóre procedury są parametryzowane. Umożliwia to wstawianie do tekstu programu użytkowego nazw własnych i wielkości podanych na kartach parametrycznych.

Zbiór ZTZ podzielony jest na części, których początki i końce oznaczone są odrębnymi identyfikatorami. Poszczególne segmenty mogą być opuszczone lub włączone do programu wynikowego w zależności od podanych parametrów. Proces ten nosi nazwę generowania programów użytkowych.

Programy użytkowe uwzględniają wymagania użytkownika, a także zakres informacji, strukturę rekordów w kategoriach oraz konfigurację komputera. Z

Mgr inż. Janusz GRZEMOWSKI (ur. 1947) — ukończył studia wyższe na Wydziale Elektroniki Akademii Górniczo-Hutniczej w Krakowie — specjalność automatyka (1972). Jest asystentem w Zakładzie Automatyzacji Zarządzania Instytutu Maszyn Matematycznych (Oddział Śląski). Pracuje nad oprogramowaniem użytkowym EMC Jednolitego Systemu.

punktu widzenia składu oraz spełnianych funkcji można wśród nich wyróżnić następujące grupy:

— moduł główny
— moduły organizacji kartotek bazowych
— moduły wejścia/wyjścia.

● Moduł główny steruje przetwarzaniem, wprowadza informacje z kart, kontroluje ich kolejność oraz analizuje kody informacji wejściowych i na ich podstawie steruje przetwarzaniem. Ponadto formuje on zapisy, wprowadza do nich zmiany oraz wyprowadza drukowane informacje wyjściowe.

Ponadto moduł główny zawiera:

— makroinstrukcje definicji zbiorów wejścia/wyjścia DTF (oprócz dyskowych)

— deklaracje obszarów WE/WY dla wszystkich zbiorów (oprócz dyskowych)

— deklaracje obszarów roboczych dla zapisów kartotek bazowych

— punkty wejścia dla podprogramów organizacji zapisów dyskowych, nagłówków itp. (pisane są one przez użytkownika w języku ASSEMBLER).

● Moduły organizacji kartotek zakładają, aktualizują i reorganizują kartoteki bazowe na dyskach. Zawierają one następujące procedury:

— utrzymywania indeksów kartotek

— zakładania łańcuchów przepełnienia powstających w trakcie uzupełnień

— generowania łańcuchów adresowanych dla podstawowych powiązań między kartotekami

— kontroli wolnego miejsca w obszarach karotek na dyskach.

● Moduły wejścia/wyjścia odczytują i aktualizują zapisy dyskowe. Spełniają więc następujące funkcje:

— wyszukiwania zapisów wg klucza lub bezpośrednich adresów dyskowych

— otwierania i zamykania kartotek
— aktualizowania zapisów na dysku.

Programy wyszukiwania informacji tworzone są w celu korzystania z bazy danych powstałej w wyniku zastosowania programów użytkowych. Pobierają one zapisy lub ich grupy z kartotek, przetwarzają je i drukują w żądanych przekrojach. Funkcje, które realizują, oraz ich ilość, zależą od indywidualnych wymagań użytkowników. Pisze się je osobno dla każdego rodzaju zastosowań.

W celu ułatwienia programowania, system kartotek bazowych dysponuje zestawami makroinstrukcji dla języków ASSEMBLER, COBOL i PL-1. Makroinstrukcje te udostępniają programiście rekordy lub grupy zapisów z kartotek, deklarują obszary pamięci, prowadzą diagnostykę błędów itp. Ze względu na skomplikowaną budowę kartotek przyspiesza to znacznie pisanie programów wyszukiwania.

Kartoteki bazowe

System zawiera 4 kartoteki dyskowe: rodzajową, strukturalną, technologiczną i stanowisk pracy. Zawierają one dwa rodzaje informacji, z których część jest niezbędna do prawidłowego funkcjonowania kartotek, a pozostałe — wynikają z potrzeb i wymogów przedsiębiorstwa.

Kartoteka rodzajowa zawiera zapisy dla każdego numeru elementu występującego w kartotece strukturalnej (przez element rozumie się zespół, podzespół, część lub materiał — składnik struktury wyrobu). Numeracja elementów musi być jednoznaczna, gdyż stanowi podstawę budowy kartoteki. Kartoteka, umieszczona na dyskach, ma organizację listową z możliwością dostępu swobodnego i przewiduje obszary przepełnienia na zapis uzupełnień i zmian.

Kartoteka strukturalna zawiera opis struktury wyrobów. Jej zapisy mają postać plików jednopoziomowych zespołów (opisujących jeden poziom rozwinięcia). Pliki mogą być rozmieszczone w kartotece w dowolnej kolejności z tym jednak, że ułożenie zapisów w pliku musi być zgodne z kolejnością składników. Zapis opisujący strukturę wyrobu powinien zawierać następujące elementy: numer części głównej, numer części składowej oraz ich ilość przypadającą na jeden zespół. Kartoteka ma budowę łańcuchową. Poszczególne zapisy powiązane są za pomocą adresów dyskowych w łańcuchy („gdzie użyty" i „z czego się składa") umożliwiające szybkie poszukiwanie grup zapisów w kartotece. Proces łańcuchowania odbywa się automatycznie w czasie zakładania, aktualizacji i reorganizacji kartotek.

Kartoteka technologiczna zawiera opis technologii wykonania (montażu) elementów występujących w kartotece rodzajowej. Każdy jej zapis powinien zawierać: numer części, numer stanowiska roboczego i numer kolejny operacji technologicznej. Kartoteka składa się z plików zapisów opisujących technologię wykonania jednej części. Ma ona budowę łańcuchową. W łańcuchy powiązane są tutaj operacje technologiczne występujące w ramach poszczególnych plików.

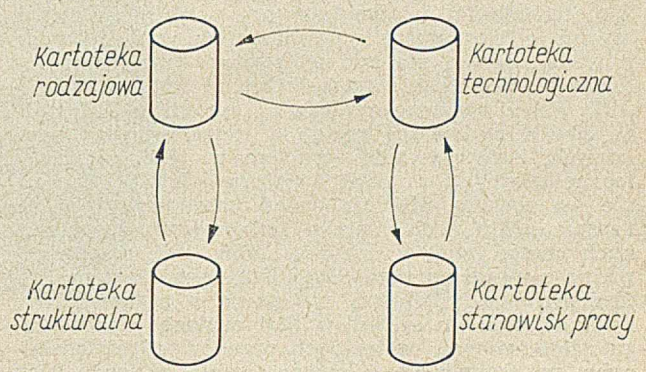
Kartoteka stanowisk pracy zawiera zapisy dotyczące funduszu czasu pracy i obciążenia dla każdego jednoznacznie określonego stanowiska roboczego. Informacją wymaganą przez system jest numer stanowiska roboczego. Kartoteka stanowisk pracy ma analogiczną organizację, jak kartoteka rodzajowa.

Powiązania między kartotekami

Dane dotyczące jednego elementu rozrzucone są we wszystkich kartotekach. W celu ułatwienia zebrania całości informacji odnośnie danej części poszczególne zapisy połączone są indeksami. I tak na przykład zapisy w kartotece rodzajowej zawierają adresy zapisów opisujących strukturę i technologię wykonania i odwrotnie.

Schematyczne powiązania kartotek przedstawia rys. 2.

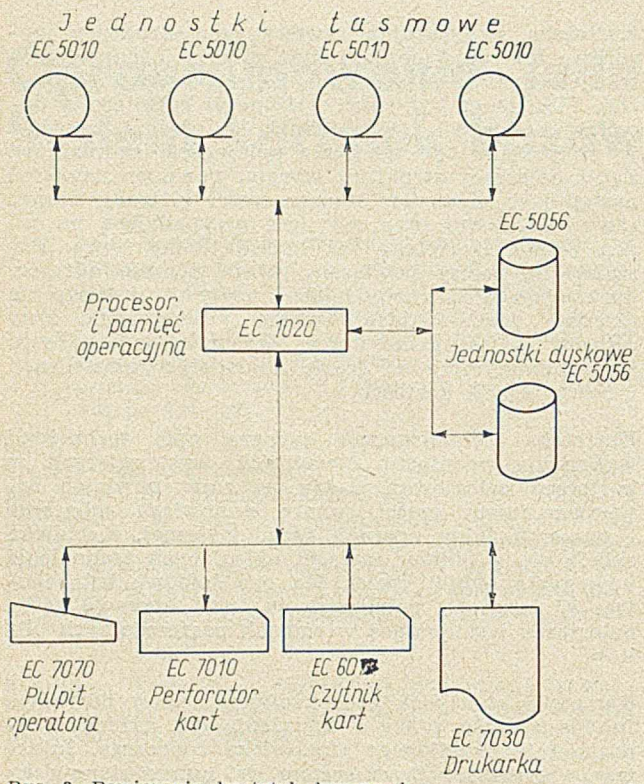
Dostęp do informacji zawartych w kartotekach bazowych jest możliwy jedynie poprzez kartoteki o dostępie swobodnym. Dzięki powiązaniom adresowym dostęp do informacji o strukturze i technologii poprzez kartotekę rodzajową lub stanowisk pracy jest łatwy i szybki.



Rys. 2. Wymagana konfiguracja komputera

Wymagana konfiguracja maszyny

System może być wykorzystany na maszynach Jednolitego Systemu pracujących pod kontrolą DOS/JS i posiadających więcej niż 48 KB pamięci operacyjnej oraz minimum 2 jednostki dyskowe po 7,25 MB.



Rys. 3. Powiązania kartotek bazowych

Pełną wymaganą konfigurację komputera przedstawia rys. 3.

Możliwości wdrożenia systemu

System kartotek bazowych został opracowany dla przemysłu maszynowego, jednakże w związku z możliwością łatwego dostosowania go do różnorodnych wymagań, może również znaleźć zastosowanie w innych gałęziach gospodarki narodowej, zwłaszcza zaś wszędzie tam, gdzie produkcja ma charakter seryjny lub małoseryjny z operacjami montażu.

Podstawowym warunkiem wdrożenia systemu jest posiadanie ujednoliconej dokumentacji konstrukcyjnej i technologicznej. Każdy element, operacja itp. musi mieć określony w jednoznaczny sposób indeks. Może on być na przykład indeksem przeznaczonym tylko do organizacji systemu przetwarzania, niezależnym od aktualnie używanego w przedsiębiorstwie.

Ponieważ system kartotek bazowych jest odpowiednikiem pakietu Bill of Material Procesor (BOMP) dla IBM-360, przedsiębiorstwa wykorzystujące system BOMP na maszynach IBM mogą w stosunkowo łatwy sposób, wykorzystując opisany system kartotek bazowych, przenieść swoje zbiory i programy wyszukiwania na komputer Jednolitego Systemu.

Literatura

System 360 Bill of Material Processor

[1] Programmer's Manual
[2] Operator's Manual
[3] A Maintenance and Retrieval System
[4] Application Description.

Konferencja: „Selektywna Dystrybucja Informacji"

Jedną z form rozpowszechniania informacji naukowej, technicznej i organizacyjnej z wykorzystaniem maszyn cyfrowych, co z powodzeniem jest stosowane w wielu krajach, stanowi selektywna dystrybucja informacji (*Selective Dissemination of Information* — SDI). Dzięki sprawnej organizacji systemów SDI można śledzić na bieżąco postęp w zakresie badań naukowych, rozwiązywanie problemów technicznych i technologicznych w skali światowej. Istotne znaczenie ma również SDI dla badań prognostycznych. W naszym kraju problematyka zastosowania nowoczesnych metod obsługi użytkowników jest szczególnie ważna z uwagi na organizację krajowego systemu informacyjnego. Problemy związane z organizacją i funkcjonowaniem systemów SDI znalazły się w kręgu zainteresowań centralnych organów odpowiedzialnych za organizację i wypracowanie nowych form działalności informacyjnej.

W dniach od 10 do 13 września 1974 r. odbyła się w Polanicy-Zdroju I Krajowa Konferencja poświęcona selektywnej dystrybucji informacji. Organizatorami Konferencji są: Centrum Informacji Naukowo-Technicznej oraz Biblioteka Główna i Ośrodek Informacji Naukowo-Technicznej Politechniki Wrocławskiej.
Głównym celem Konferencji jest zapoznanie uczestników z organizacją systemów selektywnej dystrybucji informacji, omówienie wybranych systemów SDI o zasięgu międzynarodowym oraz przedyskutowanie problemów wykorzystania SDI w krajowym systemie INTE.
Wszelkich informacji udziela Sekretariat Konferencji: Biblioteka Główna i Ośrodek Informacji Naukowo-Technicznej Politechniki Wrocławskiej, 50—370 Wrocław, ul. Wybrzeże Wyspiańskiego 27, pokój 255, tel. 22-70-51 w. 903.

Informatyka w liceach

Prezydium Rządu Decyzją Nr 3/74 zobowiązało Ministerstwo Nauki Szkolnictwa Wyższego i Techniki oraz Ministerstwo Oświaty i Wychowania do opracowania programu kształcenia i szkolenia kadr dla informatyki. Ogromna rola przypadła Ministerstwu Oświaty i Wychowania, które zostało zobowiązane do kształcenia kadry informatyków na poziomie średnim. Kształcenie to ma przyjąć dwie formy:
— wprowadzenie przedmiotu „podstawy informatyki" do liceów ogólnokształcących jako elementu wykształcenia ogólnego
— kształcenie specjalistyczne w 2-letnich pomaturalnych szkołach specjalistycznych.

Bliższego wyjaśnienia wymaga pierwszy problem, który już w bieżącym roku szkolnym 1974/75 zostanie wdrożony tytułem eksperymentu w około 20-tu szkołach średnich ogólnokształcących. Eksperyment jest prowadzony dwoma programami i nadzorowany przez dwie instytucje: Instytut Kształcenia Nauczycieli (I program — o charakterze informatycznym) oraz Instytut Programów Szkolnych Ministerstwa Oświaty i Wychowania*) (II program — o charakterze ogólnym).

Szczegółowa prezentacja I-go programu zostanie przedstawiona przy innej okazji. Poniżej natomiast zostanie omówiony II-gi program. **Jego autorzy postawili sobie za cel, aby poprzez wprowadzenie „podstaw informatyki" do liceów, stworzyć określoną „kulturę informatyczną" wśród uczniów, kulturę pozwalającą im rozumieć wiele trudnych problemów z innych przedmiotów i przygotowującą ich do przyjęcia w przyszłości większej sumy wiadomości z tego zakresu. Uczniowie nie mają nauczyć się informatyki jako zawodu, a jedynie poznać ją w sensie ogólnym.** Podobne formy kształcenia w szkołach średnich zostały z powodzeniem wprowadzone we Francji, Kanadzie, Anglii, RFN itp.

Autorzy II-go programu założyli, że „podstawy informatyki" będą realizowane w III-ch klasach liceów ogólnokształcących w wymiarze dwóch godzin w ciągu tygodnia. Tak skromna ilość godzin świadczy o tym, że uczniowie otrzymują tylko pojęciowe podstawy informatyki oraz zdobędą elementarne umiejętności w zakresie rozwiązywania problemów za pomocą elektronicznych maszyn cyfrowych. Dalszą, bardziej specjalistyczną wiedzę w tym zakresie absolwenci liceów otrzymają w szkołach pomaturalnych lub na wyższych uczelniach.

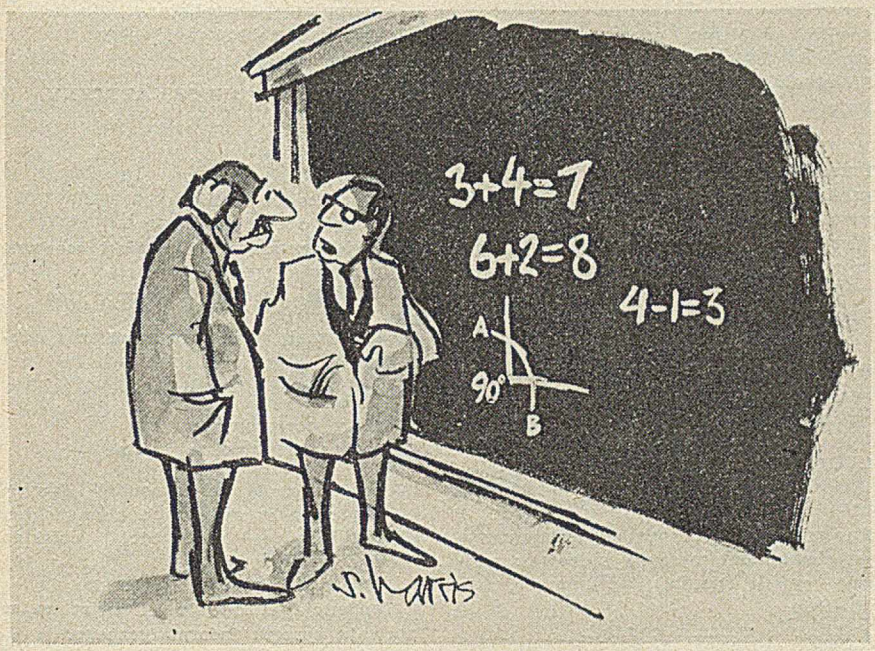
W omawianym programie przewiduje się wprowadzenie ucznia w dziedzinę informatyki poprzez następujące zagadnienia:

1. Przesłanki rozwoju informatyki i przetwarzania danych
2. Rys historyczny — rozwój urządzeń informatyki
3. Cybernetyczne podstawy informatyki
4. Rozwiązywanie problemów — algorytmy
5. Organizacja elektronicznej maszyny cyfrowej
6. Elementy programowania w języku FORTRAN
7. Sprzęt informatyczny
8. Technologia przetwarzania
— organizacyjne przygotowanie użytkownika
— projektowanie systemów EPD
— organizacja ośrodka obliczeniowego
9. Przegląd zastosowań informatyki
10. Ćwiczenia

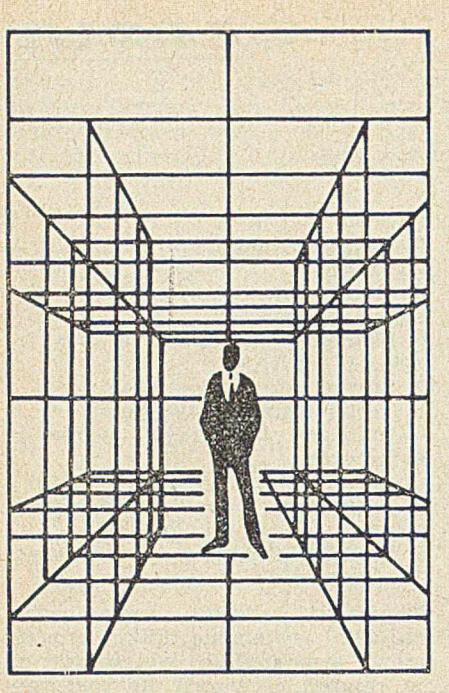
Do zadań każdego ucznia będzie należało wykonanie kilku programów, samodzielnie wytestowanych w ramach ćwiczeń na maszynie cyfrowej. **Zakres programu jest pomyślany tak, aby mógł go opracować każdy przeciętny uczeń. Jednak trwałe przyswojenie materiału nauczania w dużej mierze zależy od aktywnego udziału ucznia w lekcji, od jego zainteresowania tematem. Treść programu dostarcza uczniowi nie tylko nowych wiadomości, lecz umożliwia również dostrzeganie związków logicznych między odległymi na pozór dziedzinami.**

Ministerstwo Oświaty i Wychowania zdecydowało, że wykładowcami II-go programu będą bardzo dobrzy fachowcy — informatycy z ośrodków obliczeniowych mający niezbędne doświadczenie pedagogiczne. Licea ogólnokształcące jeszcze nie dysponują wykwalifikowaną kadrą etatowych nauczycieli informatyki. Już w bieżącym roku Instytut Kształcenia Nauczycieli przystąpił do realizacji szerokiego programu kształcenia nauczycieli matematyki i fizyki w zakresie informatyki. Należy sądzić, że już w roku następnym funkcję nauczycieli dochodzących przejmą nauczyciele etatowi.

W celu prawidłowej realizacji całego eksperymentu Ministerstwo Oświaty i Wychowania postanowiło przygotować niezbędne pomoce naukowo-dydaktyczne. Uczniowie będą otrzymywać w ciągu roku zeszyty skryptu systematycznie dostarczanego w miarę realizacji programu. Wszystkie szkoły otrzymają duże zdjęcia poszczególnych elementów komputera. Postanowiono również wypożyczyć z Instytutu Brytyjskiego w Warszawie kolorowe filmy o komputerach w wersji polskiej. Wykładowcy są zobowiązani do skompletowania takich pomocy naukowych, jakie są dostępne w ośrodkach obliczeniowych, a jednocześnie niezbędne



*) Przy stałej współpracy Instytutu Badań Pedagogicznych



w procesie nauczania „podstaw informatyki". Do pomocy tych należy zaliczyć:
— karty i taśmy perforowane
— taśmy magnetyczne z zapisem ziarnistym
— obwody drukowane i scalone
— komunikaty z monitora
— listę rozkazów w języku FORTRAN
— tabulogramy użytkowe
— polską normę pojęć informatycznych
— wzorce dokumentacji projektowo-programowej
— wzorce dokumentów źródłowych
— schematy budowy maszynowych nośników informacji
— schematy budowy kodów, symboli, indeksów
— schematy blokowe programów
— schematy blokowe przebiegów działań na EMC
— typowe algorytmy, zastosowania itp.

Zestaw ten jest tylko częścią niezbędnych pomocy naukowych stosowanych w nauczaniu „podstaw informatyki". Dalsze pomoce będą w następnych latach kompletowane.

Ogromnie istotnym elementem w procesie uczenia tego przedmiotu jest bezpośredni dostęp do komputera i korzystanie z niego. Uczniowie będą mieli taką okazję w dwóch przypadkach: na początku nauczania przedmiotu odbędą wycieczkę do ośrodka obliczeniowego, a pod koniec nauczania — będą testować na komputerze napisane przez siebie programy. W tym celu Ministerstwo Oświaty i Wychowania wydzierżawi dla każdej klasy około 20 godz. pracy maszyny.

Mamy nadzieję, że dyrekcje szkół i wykładowcy dołożą wszelkich starań, aby eksperyment wdrożenia przedmiotu „postawy informatyki" zakończył się pełnym powodzeniem. Natomiast uczniowie postarają się opanować tak materiał aby na świadectwach były same piątki tym bardziej, że przedmiot jest obowiązkowy. **Na wszystkich kierunkach studiów będzie w najbliższym czasie wykładana informatyka. Tak więc wiedza zdobyta w szkole średniej w znacznym stopniu ułatwi każdemu uczniowi — przyszłemu studentowi — pogłębianie i wykorzystanie wiedzy informatycznej.**

Bogumił Stachura
Ministerstwo Oświaty i Wychowania

Tematyka programu i materiał nauczania

I. **PRZESŁANKI ROZWOJU INFORMATYKI I PRZETWARZANIA DANYCH** (2 godz)

1. Znaczenie informacji w życiu społeczno-gospodarczym. Współczesne źródła danych i konieczność ich przetwarzania.
2. Efekty rozwoju informatyki i przetwarzania danych.

II. **ROZWÓJ URZĄDZEŃ INFORMATYKI — RYS HISTORYCZNY** (1 godz.)

1. Od Pascala do Holleritha
2. Generacje maszyn cyfrowych
3. Rozwój i eksploatacja maszyn cyfrowych w Polsce.

III. **CYBERNETYCZNE PODSTAWY INFORMATYKI** (2 godz.)

1. Przedmiot cybernetyki
2. Pojęcie układu, układu względnie odosobnionego
3. Układ informacyjny — sieć informacyjna
4. Układ cybernetyczny (maszyna cybernetyczna)
5. Informatyka jako dział cybernetyki stosowanej.

IV. **ROZWIĄZYWANIE PROBLEMÓW — ALGORYTMY** (15 godz.)

1. Rozwiązywanie problemów:
Rozwiązywanie problemów jako przetwarzanie informacji

Analiza problemu, operacje składowe i ustalenie ich wzajemnych powiązań.
Obliczenia, rodzaje i opisy obliczeń.
Pojęcie algorytmu.

2. Algorytmy:
Przedstawienie algorytmu, słowne i graficzne reprezentacje.
Sieć działań.
Wyróżnienie cech i rodzajów algorytmów na podstawie wybranych przykładów.
Zapis algorytmów w różnych językach.

V. **ORGANIZACJA ELEKTRONICZNEJ MASZYNY CYFROWEJ** (łącznie z pkt. VI — 30 godz.)

1. Ogólny zapis maszyny cyfrowej:
Budowa maszyny cyfrowej — schemat blokowy.
Funkcje i związki między podstawowymi elementami maszyny cyfrowej: wejście—wyjście, pamięć, arytmometr, sterowanie.
Funkcje i zadania pamięci.

2. Arytmetyka maszyn cyfrowych:
Systemy liczbowe.
Operacje arytmetyczne.

3. Podstawowy cykl pracy EMC:
Przedstawienie danych.
Rozkaz i jego budowa.
Adresowość maszyn cyfrowych.
Adresy rozkazów.
Rozkazy adresowane sekwencyjnie i niesekwencyjnie.
Ogólny cykl pracy EMC.

4. Hipotetyczny model maszyny cyfrowej:
Rozkazy wejścia i wyjścia.
Rozkazy przesłania słów.
Rozkazy arytmetyczne.
Rozkazy podejmowania decyzji.
Język wewnętrzny maszyny. Pojęcie programu dla EMC.

5. Zastosowanie poznanych rozkazów do programowania.

6. Język symboliczny maszyny cyfrowej.
Przykłady programów w języku symbolicznym modelu EMC.

7. Język symboliczny a język wewnętrzny.
Konieczność tłumaczenia; translatory.

VI. **ELEMENTY PROGRAMOWANIA W JĘZYKU FORTRAN**

1. Języki zorientowane problemowo i maszynowo
2. Struktura języków algorytmicznych
3. Ogólna struktura programów w języku FORTRAN
4. Zmienne i ich typy
5. Funkcje standardowe, wyrażenia arytmetyczne i logiczne
6. Instrukcje podstawienia
7. Instrukcje sterujące
8. Procedury
9. Program główny
10. Instrukcje wejścia—wyjścia
11. Segmenty programu MASTER, SUBROUTINE, FUNCTION
12. Idea translacji
13. Kompilatory FORTRAN-u, uruchamianie programów:
— wiersze sterujące kompilacją
— komunikaty kompilatora
— sygnalizacja błędów

VII. **SPRZĘT INFORMATYCZNY** (4 godz.)

1. Budowa EMC:

— jednostka centralna
— pamięć operacyjna
— pamięć zewnętrzna (pamięci masowe)
— urządzenia wejścia—wyjścia
— pozostałe urządzenia peryferyjne

2. Urządzenia do przygotowania danych:

— kartowe
— taśmy perforowane
— taśmy magnetyczne i inne

3. Urządzenia do teletransmisji danych
4. Urządzenia w systemach abonenckich
5. Produkcja i eksploatacja polskich maszyn cyfrowych
6. Jednolity System EMC RIAD
7. Tendencje europejskie i światowe w rozwoju sprzętu informatycznego.

VIII. TECHNOLOGIA PRZETWARZANIA DANYCH (6 godz.)

1. Organizacyjne przygotowanie użytkowników:

— strumienie informacyjne w obiektach gospodarczych i innych
— zasady przebudowy nośników informacji, kodów, indeksów, normatywów, dokumentów itp.

2. Projektowanie systemów EPD:
— etapy projektowania systemów i tematów
— schemat blokowy przetwarzania
— założenia do programów.

3. Organizacja ośrodków informatycznych:

— charakterystyka sieci ośrodków obliczeniowych (zakładowych, branżowych, usługowych, uczelnianych)
— oragnizacja pracy typowego ośrodka usługowego (przygotowanie, wdrożenie i eksploatacja)
— organizacja pracy ośrodka zakładowego.

IX. PRZEGLĄD ZASTOSOWAŃ INFORMATYKI (4 godz.)

1. Zastosowanie EMC w modelach symulacyjnych
2. Zastosowanie EMC w obliczeniach numerycznych
3. Zastosowanie EMC w optymalizacji
4. Zastosowanie EMC w zarządzaniu przedsiębiorstwem:

— systemy ewidencyjne (gosp. mat., płace)
— systemy planistyczne (planowanie produkcji)

5. Zastosowanie EMC w systemach państwowych:

— statystyka (SPIS)
— ewidencja (EWIKAN, MAGISTER)
— planowanie (CENPLAN).

X. ĆWICZENIA (6 godz.)

1. Zwiedzanie ośrodka obliczeniowego
2. Pisanie prostego programu w języku FORTRAN i uruchomienie go na EMC. (20 godzin na klasę, 0,5 godz. na ucznia).

Ośrodek Obliczeniowy Akademii Rolniczej w Poznaniu

W maju 1971 r. powołano w Akademii Rolniczej w Poznaniu Zakład Metod Matematycznych i Statystycznych. Jest to jednostka międzywydziałowa Akademii podległa bezpośrednio Rektorowi. Zakładowi postawiono następujące zadania:

- prowadzenie zajęć dydaktycznych w zakresie przedmiotów obejmujących zastosowania matematyki, statystyki matematycznej i elektronicznej techniki obliczeniowej
- prowadzenie badań naukowych w zakresie metod matematycznych i statystycznych, przydatnych w nauce i praktyce rolniczej
- udzielanie usług jednostkom naukowo-badawczym uczelni w zakresie stosowania metod matematycznych i statystycznych oraz użytkowania elektronicznych maszyn cyfrowych
- kształcenie kadry naukowej do twórczego rozwijania metod matematycznych i statystycznych oraz elektronicznej techniki obliczeniowej.

Dla umożliwienia realizacji tych zadań utworzono w ramach Zakładu Ośrodek Obliczeniowy, wyposażony w komputer ODRA 1204 w następującym zestawie:

— jednostka centralna z pamięcią 16K
— czytnik taśmy papierowej CT 1001 — 3 szt
— perforator taśmy papierowej D 102 — 2 szt
— pamięć bębnowa PB 204: 2 zestawy bębnów + jednostka sterująca.

Urządzenia peryferyjne stanowią: automat piszący OPTIMA 527 (6 szt), automat organizacyjny OPTIMA 528 (1 szt), dalekopis LORENZ LC133 (2 szt).

Obecnie w Ośrodku na stanowiskach technicznych zatrudnionych jest 16 osób, w tym 1 inżynier elektronik, 3 technicy elektronicy, 2 operatorzy ODRY 1204, 9 operatorów urządzeń peryferyjnych, 1 pracownik przyjmujący i wydający zlecenia.

Oprogramowanie podstawowe ODRY 1204 zawiera:

— systemy operacyjne: SOW, BOSS, MASON
— translatory języków programowania: JAS, MOST, ALGOL
— fabryczny zestaw programów, podprogramów i procedur w wyżej wymienionych językach.

W ciągu dwóch lat istnienia Ośrodka zdołano zgromadzić dodatkowo 40 programów i procedur dotyczących:

— jedno i wielozmiennej analizy wariancji dla różnych układów doświadczalnych
— różnych zagadnień analizy regresji
— metod identyfikacyjnych
— metod taksonomicznych
— analizy składowych głównych itp.

Opracowywaniem wyżej wymienionych programów i procedur zajmowali się pracownicy naukowo-dydaktyczni i naukowo-badawczy Zakładu Metod Matematycznych i Statystycznych oraz współpracujący z nimi matematycy z Instytutu Matematyki UAM, Zespołu Matematyki AR i Zakładu Genetyki Roślin PAN.

Część wymienionych programów i procedur opublikowano już w Rocznikach Akademii Rolniczej w Poznaniu (seria ALGORYTMY BIOMETRYCZNE I STATYSTYCZNE, Zeszyt 1 i 2). Pozostałe będą opublikowane w następnych zeszytach.

W związku ze wzrostem zadań obliczeniowych przewiduje się w bieżącym roku rozbudowę sprzętu komputerowego o dwie jednostki pamięci bębnowej i drukarkę wierszową do ODRY 1204.

Przewiduje się też zorganizowanie filii abonenckiej Środowiskowego Laboratorium Przetwarzania Danych (do roku 1975). Dalsze plany przewidują uzyskanie maszyny cyfrowej trzeciej generacji (do roku 1978).

Z usług ośrodka — poza Instytutami Akademii Rolniczej korzystają: — Zakład Genetyki Roślin PAN, Instytut Ochrony Roślin, Instytut Hodowli i Aklimatyzacji Roślin, Instytut Krajowych Włókien Naturalnych, Centralne Laboratorium Drobiarstwa, Kombinat Ogrodniczy Naramowice, Zakład Dendrologii PAN w Kórniku, Szczecińskie Przedsiębiorstwo Miernicze, Nadodrzańskie Zakłady POW w Dębnie Lubuskim, WZPGR Zielona Góra, Akademia Medyczna, MIASTOPROJEKT, ETOB, ENERGOPROJEKT, TASKOPROJEKT, BIPROMASZ, CUKROPROJEKT i inne instytucje.

Hanna Chudzik
Stanisław Czajka
Akademia Rolnicza
Poznań

Kartka z Kielc

Na terenie Kielecczyzny działa od 1971 r. Wojewódzka Komisja Koordynacji Terenowo-Branżowej Informatyki, której członkami są przedstawiciele ponad 20 ośrodków obliczeniowych największych zakładów pracy województwa kieleckiego.

Komisja zajmuje się oceną stanu przygotowania i wprowadzania systemów informatycznych oraz opracowywaniem programów rozwoju informatyki w różnych branżach województwa. Komisja popularyzuje informatykę i czuwa nad szkoleniem kadry. Między innymi z inicjatywy Komisji przeprowadzono wiele kursów programowania i konserwacji sprzętu, kursy dla kadry kierowniczej i spotkania fachowców. W tym zakresie Komisja opiera się na działalności ZETO-Kielce.

Z inicjatywy działaczy kieleckich Kuratorium Okręgu Szkolnego otworzyło 1 klasę Pomaturalnego Studium Przetwarzania Danych i 1 klasę Pomaturalnego Studium Programowania EMC.

W przedsiębiorstwach województwa kieleckiego spory potencjał obliczeniowy stanowią jeszcze maszyny licząco-analityczne. W końcu 1973 r. było zainstalowanych 26 zestawów MLA w około 10 największych zakładach przemysłowych. Komputerów do przetwarzania danych jest bardzo mało. Do niedawna dysponowały nimi: FSC Starachowice, ZEOW Radom, Huta Ostrowiec, a ponadto GUS-Radom.

Kadra, pracująca w przedsiębiorstwach stosujących technikę obliczeniową, składa się przede wszystkim z projektantów i koordynatorów systemów. Jest jeszcze bardzo nieliczna. Natomiast w ośrodkach obliczeniowych (w tym również MLA) sytuacja przedstawia się nieco lepiej, gdyż pracuje tam ponad 200 projektantów i programistów, prawie 100 konserwatorów sprzętu, ok. 300 operatorów — w sumie ok. 900 osób. Ocenia się, że do 1976 r. trzeba będzie przeszkolić co najmniej 2500 osób. Duże zadania w tym zakresie ma WSI Kielce, która niestety dysponuje tylko przestarzałą maszyną ODRA 1013.

ZETO-Kielce, utworzone w 1968 r., przez dłuższy czas praktycznie nie dysponowało dobrym sprzętem. Korzystało wtedy z pomocy ZETO-Katowice. Dopiero w 1972 r. otrzymało ODRĘ 1304 oraz urządzenia do przygotowania danych i mogło przystąpić do właściwych usług informatycznych na swoim terenie. Dzięki pomocy władz wojewódzkich uzyskano budynek, który odpowiednio dostosowano do tej działalności. W końcu 1973 r. zatrudniano tam 120 pracowników, w tym 35 projektantów i programistów. Wartość usług wzrosła do kilkunastu milionów złotych. Zwiększyła się do kilkudziesięciu liczba opracowanych systemów dla różnych użytkowników i obecnie już eksploatowanych.

Rozpoczęto budowę nowego obiektu, który ma być zakończony w połowie 1975 r. W tym okresie wzrośnie też znacznie moc obliczeniowa, gdyż mają tu pracować 3 duże komputery. Planuje się dalszy rozwój ZETO w Kielcach oraz zorganizowanie oddziału w Radomiu.

ZETO-Kielce opracowało i oddało do eksploatacji kilkanaście systemów przetwarzania danych na komputerze ODRA 1304. Dotyczą one głównie ewidencji gospodarki materiałowej. Przykładami są tu następujące systemy:

— **Dla Kieleckiego Przedsiębiorstwa Robót Mostowych:** system, opracowany przez Annę Zarębę, Janusza Sliwińskiego, Janinę Nawierską, Henryka Oleszczaka i Zygmunta Dębskiego, w języku COBOL, umożliwia ewidencję obrotów i rozliczanie zużycia materiałów. Koszt dokumentacji projektowo-programowej wyniósł 495 tys. zł, testowania i uruchamiania programów — 131,5 tys. zł, wdrożenia i eksploatacji próbnej — 387 tys. zł. Eksploatacja systemu kosztuje 50 tys. zł mies. Może on być wykorzystany przez inne przedsiębiorstwa budowlane podległe Ministerstwu Komunikacji.

— **Dla Przedsiębiorstwa Geologicznego** w Kielcach:

System, opracowany przez Urszulę Kusto, Stanisława Polańskiego i Janusza Sliwińskiego w języku PLAN, służy również do ewidencji obrotów i rozliczania zużycia materiałów. Koszt dokumentacji projektowo-programowej wyniósł 300 tys. zł, testowania i uruchamiania programów — 142 tys. zł, wdrażania i eksploatacji próbnej — 472,5 tys. Eksploatacja ciągła systemu kosztuje 33 tys. zł. Może on być wykorzystywany przez inne przedsiębiorstwa geologiczne w kraju.

— **Dla Zakładów Przemysłu Dziewiarskiego REKORD** w Jędrzejowie: System opracował Zespół w składzie: Maria Zimech-Burgel, Janusz Śliwiński, Stanisław Polański, Izabella Kowalik, Janusz Mazur. Służy od do prowadzenia ewidencji stanów i obrotów magazynowych. Koszty były następujące:

dokumentacja projektowo-programowa — 420 tys. zł
testowanie i uruchamianie programów — 83 tys. zł

eksploatacja ciągła kosztuje 9,5 tys. zł.

— **Dla Przedsiębiorstwa Budowlano-Montażowego** ORGANIKA w Pionkach: Zespół w składzie: Anna Zglińska, Izabella Kowalik, Janina Mazur, Eligiusz Grądzki — opracował system ewidencji stanów, obrotów i zużycia materiałów. Koszty przygotowania systemu wyniosły ok. 650 tys. zł.

— **Dla Koneckich Zakładów Odlewniczych** w Końskich opracowano system ewidencji materiałowej kosztem ok. 600 tys. zł. Autorami systemu są: Anna Łoboda, Anna Wysocka i Ryszard Liguz.

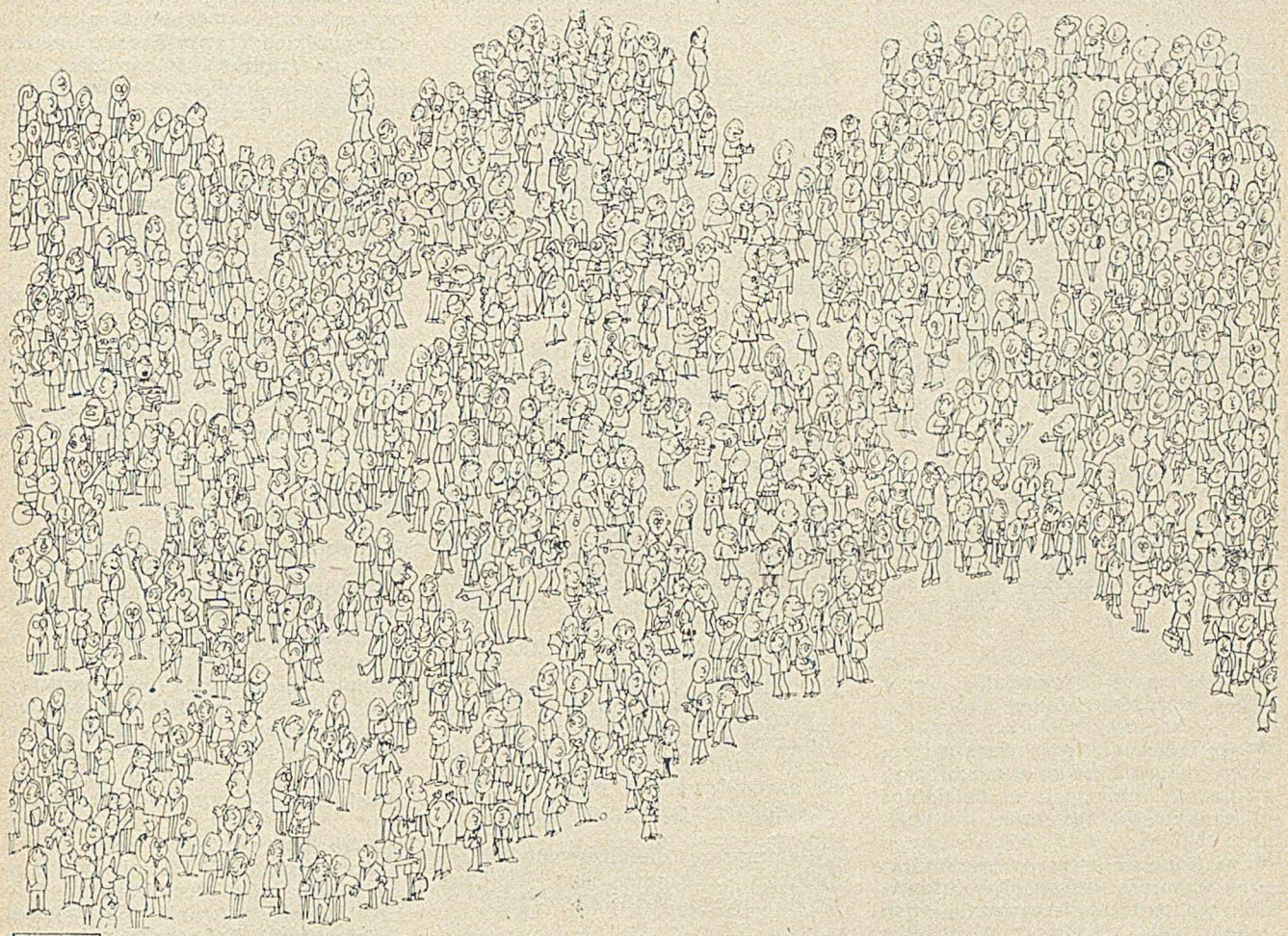
Inny charakter ma opracowany dla **Generalnego Realizatora Inwestycji** w Nowinach system sterowania procesem inwestycyjnym (PERT); autorami są Tomasz Stanaszek i Stanisław Urbanek.

Dla **Centralnego Związku Spółdzielczości Pracy** Anna Wysocka i Stanisław Urbanek opracowali analizę opłacalności dewizowej dostaw eksportowych. Koszt wyniósł około 65 tys. złotych.

Dla **Kombinatu Przemysłu Łożysk Tocznych** w Kielcach opracowano zespół programów do obliczenia katalogu normatywnych kosztów bezpośrednich. Autorami są: Danuta Bakalarczyk, Janina Mazur, Józef Rola.

Pracownicy ZETO Kielce opracowują i wdrażają szereg systemów dla takich użytkowników, jak Zakłady Mięsne (rozliczenie produkcji i sprzedaży), Przedsiębiorstwo Hydrogeologiczne w Krakowie (gospodarka materiałowa) i inne.

Na zlecenie OBRI opracowuje się na ODRĘ 1304 system ewidencji kadr nauczycielskich w Polsce. W skład zespołu autorskiego wchodzą: B. Maciejewski, M. Nowakowski, S. Polański, B. Stachura, S. Urbanek, A. Zaręba. Preliminowany koszt systemu ponad 1 mln. zł.



IFIP CONGRESS 74

INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING

AUGUST 5–10 STOCKHOLM SWEDEN

IFIP to skrót nazwy International Federation for Information Processing (Międzynarodowa Federacja do Spraw Przetwarzania Informacji), stowarzyszenia powstałego i znajdującego się pod patronatem UNESCO (Organizacji Narodów Zjednoczonych do Spraw Oświaty, Nauki i Kultury). Idea powołania tego rodzaju międzynarodowego zrzeszenia narodowych stowarzyszeń informatyków wpłynęła na początku 1958 roku, a więc zaledwie w 7 lat po zastosowaniu pierwszego komputera do celów przetwarzania danych gospodarczych. Inicjatorem tej idei był znany amerykański ekspert w dziedzinie informatyki Isaac Auerbach (wydawca słynnego wydawnictwa encyklopedycznego w zakresie sprzętu, oprogramowania i zastosowań komputerów), który — działając z ramienia amerykańskiego zrzeszenia informatyków (Joint Computer Committee) — wystosował do ówczesnego przewodniczącego UNESCO prof. Pierre Auger'a memoriał na temat konieczności rozwinięcia szerokiej współpracy międzynarodowej w dziedzinie informatyki, a zwłaszcza szybkiego zwołania odpowiedniej konferencji naukowej.

W czerwcu 1959 roku w paryskim Pałacu Narodów, ówczesnej siedzibie UNESCO, odbyła się pierwsza międzynarodowa konferencja naukowa w dziedzinie informatyki (oficjalna nazwa: International Conference on Information Processing) z udziałem ponad 1800 delegatów z 37 krajów. Jednocześnie w salach wystawowych paryskiego Grand Palais czynna była wystawa sprzętu informatycznego prezentująca całość imponujących już wówczas osiągnięć technicznych w tej dziedzinie (pierwsze komputery półprzewodnikowe). W czasie konferencji wstępnie sformułowano strukturę organizacyjną i cele IFIP; projekt statutu został ratyfikowany przez 13 stowarzyszeń narodowych. Według oficjalnej definicji IFIP jest federacją narodowych stowarzyszeń naukowo-technicznych (lub ich sekcji) obejmujących swą działalnością problematykę przetwarzania informacji.

W 14 lat po powstaniu Federacji jej członkami jest już 35 stowarzyszeń narodowych. Polska przystąpiła oficjalnie do Federacji w 1961 r. i jest w niej reprezentowana przez Polską Akademię Nauk. Aktualnym przedstawicielem jest prof. L. Łukaszewicz.

Podstawowym ogólnym celem działalności IFIP jest skupianie naukowców i innych specjalistów z dziedziny przetwarzania informacji. Cele szczegółowe obejmują:

- *popieranie rozwoju nauki i techniki w dziedzinie informatyki*

Tabela I.

	IFIP	MEDINFO	ŁĄCZNIE
Szwecja	750	182	932
USA	373	80	453
RFN	297	97	394
Wielka Brytania	158	88	246

Tabela II.

	IFIP	MEDINFO	ŁĄCZNIE
Węgry	50	11	61
Czechosłowacja	45	5	50
ZSRR	32	7	39
NRD	16	9	25
Polska	22	0	22
Bułgaria	6	1	7
Jugosławia	5	0	5
Kuba	2	2	4
Rumunia	2	0	2
Koreańska RLD	1	0	1
	181	35	216

- *inicjowanie i rozwijanie współpracy międzynarodowej*
- *stymulowanie prac badawczych i rozwojowych oraz zastosowań przetwarzania informacji w działalności naukowej i społecznej*
- *przyśpieszanie rozpowszechniania oraz wymiany osiągnięć i doświadczeń na temat przetwarzania informacji*
- *popieranie rozwoju działalności szkoleniowej.*

Najwyższą władzą IFIP jest Walne Zgromadzenie (General Assembly) które zbiera się co roku. W jego skład wchodzi po 1 przedstawicielu każdego stowarzyszenia członkowskiego. Walne Zgromadzenie decyduje o ogólnej polityce, programie działalności, przyjmowaniu nowych członków, wyborach władz oraz budżecie Federacji. Działalnością Federacji kieruje Zarząd (Executive Body), złożony z przewodniczącego, 3 wiceprzewodniczących, sekretarza i skarbnika, których na okres trzyletni wybiera Walne Zgromadzenie. Działalność bieżącą prowadzi Sekretariat. Ostatnim wreszcie organem IFIP jest Rada Nadzorcza (Council), złożona z przewodniczącego oraz 6 wybieranych członków. Zbiera się ona dwa razy w roku i podejmuje niezbędne decyzje pomiędzy posiedzeniami Walnego Zgromadzenia. Prawną siedzibą Federacji jest Genewa. Kongresy IFIP są organizowane co 3 lata.

Łącznie z konferencją paryską w 1959 r. odbyło się już 6 kongresów IFIP: w Monachium, w Nowym Jorku (największy w historii tych imprez), w Edynburgu w Lublanie i w Sztokholmie, w bieżącym roku. Następny Kongres (IFIP 77) odbędzie się w Toronto. Każdemu Kongresowi IFIP towarzyszy wystawa sprzętu i zastosowań komputerowych.

* * *

Kongres IFIP 74 odbywał się w Sztokholmie, od 5 do 10 sierpnia br.

Organizatorowi — szwedzkiemu narodowemu stowarzyszeniu przetwarzania informacji — z pomocą finansową i organizacyjną przyszły bratnie stowarzyszenia oraz instytucje pozostałych krajów skandynawskich.

Tegoroczny, szósty już z kolei, Kongres IFIP różnił się od poprzednich tym, że oprócz tradycyjnie towarzyszącej mu wystawy sprzętu informatycznego zorganizowano drugą imprezę typu konferencji naukowej — pierwszą światową konferencję informatyki medycznej MEDINFO 74, której patronował Komitet d/s Przetwarzania Informacji Medycznej *(IFIP Committe for Medical Information Processing).*

Omówienie Konferencji MEDINFO 74, wystawy oraz bardziej interesujących wystąpień na Kongresie i Konferencji zamieścimy w następnych numerach INFORMATYKI (nasz przedstawiciel był jednym z 69 akredytowanych przy Kongresie dziennikarzy prasy fachowej).

Całą imprezę zlokalizowano na terenach sztokholmskiego centrum wystawowego MÄSSAN (Targi).

Wystawowo-halowy charakter pomieszczeń sprawiał, że — pomimo dobrej aparatury nagłośnieniowej — niektórych prelegentów słychać było nie najlepiej. Pewną wadą lokalizacji była również kilkukilometrowa odległość od centrum miasta, dolegliwość łagodzona dobrym zapleczem gastronomicznym na miejscu, ale i pogłębiona niesprawną sprzedażą biletów dojazdowych.

Oczekiwania organizatorów przeszła też frekwencja, mimo stosunkowo wysokiego kosztu uczestnictwa (ponad 100 $ USA), co prawdopodobnie spowodowało takie niedociągnięcia organizacyjne, jak brak bieżącej aktualizacji listy oraz szczegółowej statystyki uczestnictwa, nawet, na użytek prasy. Szczegółowe dane prasowe dotyczące liczby i pochodzenia uczestników kongresu i konferencji udostępnione dla prasy dotyczą stanu na dzień 30 lipca 1974 r., a więc na tydzień przed ich oficjalnym rozpoczęciem. Wynika z nich, że zarejestrowano łącznie 3836 uczestników z 50 krajów, z czego na Kongres IFIP 2954 osób, zaś na Konferencję MEDINFO 882 osób. Ponadto w piątek 9 sierpnia podano ogólną informację, że łączna liczba zarejestrowanych uczestników Kongresu IFIP wyniosła 3368 osób z 55 krajów oraz 928 osób Konferencji MEDINFO z 33 krajów, a więc łącznie 4296 osób.

Pod względem frekwencji impreza tegoroczna wysunęła się na pierwsze miejsce wśród kongresów IFIP organizowanych na terenie Europy, ustępując jedynie Kongresowi nowojorskiemu z 1965 roku (ponad 5000 uczestników). Rekordem natomiast była liczba krajów, które wydelegowały swych przedstawicieli (55).

Po raz pierwszy w historii tych kongresów zarejestrowano obecność przedstawicieli Koreańskiej Republiki Ludowej i Chin Ludowych.

Najliczniej — ze zrozumiałych względów — była reprezentowana Szwecja, skąd pochodziło ok. 25% ogólnej liczby uczestników. W szczegółach, uczestnictwo dalszych w kolejności najliczniej reprezentowanych krajów odpowiadało w zasadzie aktualnemu poziomowi ich komputeryzacji, a mianowicie: komputeryzacji (tab. I).

Zaskakującym faktem była natomiast szczególnie mała liczba uczestników z Japonii (37 osób na IFIP oraz 7 osób na MEDINFO), będąca w rażącej dysproporcji z osiągniętym w tym kraju poziomem komputeryzacji. Również stosunkowo skomnie przedstawiał się wg wspomnianych danych udział przedstawicieli krajów socjalistycznych (ok. 5,6% ogólnej liczby uczestników — tab. II).

Oprócz tego w zestawieniu uczestników figurowali liczni przedstawiciele krajów rozwijających się, np. Algerii, Arabii Saudyjskiej, Egiptu, Indii, Indonezji, Iraku, Iranu, Syjamu, czy Tanzanii. Stosunkowo licznie w porównaniu do osiągniętego poziomu komputeryzacji reprezentowane były natomiast kraje bardziej rozwinięte lecz znajdujące się w wielkiej odległości od Szwecji jak Australia (16+12), czy Republika Południowej Afryki (5++11).

Zarówno Kongresowi IFIP 74, jak i Konferencji MEDINFO 74 patronował Komitet Honorowy *(Patron Committee)* złożony z 26 reprezentantów 4 krajów skandynawskich pod przewodnictwem szwedzkiego ministra przemysłu. Zadanie przygotowania merytorycznej treści Kongresu zrealizował powołany po poprzednim kongresie 11-osobowy Komitet Programowy *(Programme Committee)* pod przewodnictwem prof. Herberta Freemana (USA). Komitet ten składał się z 4 przedstawicieli USA, 2 przedstawicieli ZSRR oraz po jednym przedstawicielu Francji, Japonii, Kanady, Polski i Wielkiej Brytanii. Reprezentantem Polski był prof. W. M. Turski.

Bezpośrednim przygotowaniem organizacyjnym Kongresu zajmował się 11-osobowy Komitet Organizacyjny *(Organizing Committee)* pod przewodnictwem P. Svenoniusa (Szwecja). Komitet ten wspierało 5

komitetów pomocniczych, złożonych podobnie, jak on, całkowicie z przedstawicieli gospodarzy (komitety d/s wystawy, propagandy i informacji, przygotowania i rozpowszechnienia wydawnictw itp.). Przygotowania organizacyjne wspierał również 9-osobowy amerykański komitet Kongresu *(US Committee for IFIP Congress 74)* oraz stały 6-osobowy Komitet IFIP d/s Organizacji Kongresów *(Congress Organization Committee)*. Zaakceptowany przez Komitet Programowy ostatyczny program Kongresu objął 41 referatów zamówionych *(invited papers)*, *174* referatów nadesłanych *(refereed papers)*, wybranych spośród prawie 700 otrzymanych, oraz 18 dyskusji panelowych *(panel discussions)*, dotyczących najbardziej aktualnych problemów rozwoju informatyki. W porównaniu do poprzedniego kongresu, Komitet Programowy postawił zmniejszyć o ok. 25% liczbę referatów nadesłanych, natomiast znacznie zwiększyć liczbę dyskusji panelowych (zarówno w Lubljanie, jak i Edynburgu było ich zaledwie 4). Wśród referatów zamówionych — 12 stanowiły godzinne referaty przeglądowe głównych problemów, natomiast pozostałe 29 — półgodzinne relacje na temat konkretnych osiągnięć naukowych.

Referaty oraz dyskusje panelowe zostały podzielone na 87 sesji specjalistycznych, zgrupowanych w 6 następujących sekcjach problemowych:

❶ sprzęt i architektura komputerów *(computer hardware and architecture)*
❷ oprogramowanie *(software)*
❸ aspekty matematyczne przetwarzania informacji *(mathematical aspects of information processing)*
❹ zastosowania naukowe i techniczne *(scientific and technological applications)*
❺ zastosowania w naukach społecznych i humanistyce *(application in the social sciences and the humanities)*
❻ systemy dla potrzeb zarządzania i administracji *(systems for management and administration)*.

W całokształcie obrad Kongresu odczuwalna była przygniatająca dominacja przedstawicieli Stanów Zjednoczonych. Dotyczyło to zarówno grupy referentów zaproszonych, gdzie było ich prawie połowę, jak i ogólnej liczby wszystkich prelegentów (91 przedstawicieli USA). Jeszcze bardziej widoczna dominacja widoczna była w dyskusjach panelowych, gdzie informatycy amerykańscy przewodniczyli 12 dyskusjom (2/3 ogólnej liczby dyskusji) oraz w sprawowaniu funkcji przewodniczących sesji (36 na ogólną liczbę 87). Zjawisko to ma niewątpliwy związek z liczbą eksploatowanych komputerów, skąd wynika odpowiednio wielka liczba osiągnięć naukowych oraz doświadczeń praktycznych kwalifikujących się do zaprezentowania na międzynarodowej konferencji naukowej. Pozostała część referatów rozłożyła się, zgodnie ze wspomnianą zasadą maksymalnej reprezentacji geograficznej, na 28 przedstawicieli innych krajów. Stąd nawet kraje tak zaawansowane w komputeryzacji, jak RFN lub Japonia reprezentowane były przez stosunkowo małą liczbę referatów (12 oraz 9). Dążenie do stworzenia możliwości wystąpień również przedstawicieli krajów bardzo mało zaawansowanych w komputeryzacji spowodowało, że większość krajów była reprezentowana przez 1 lub 2 referentów. W grupie tej znalazła się niestety również Polska, która mimo zgłoszenia dość dużej liczby referatów zaprezentowana została przez zaledwie 1 referat i to referat zamówiony (prof. J. Seidler *„Multiplexing Problems in Computer Communications"*). Stosunkowo niski poziom merytoryczny części wygłoszonych referatów nasuwać może wątpliwości, czy odrzucenie wszystkich nadesłanych polskich referatów było całkowicie obiektywne. Uwaga ta odnosi się również do innych krajów socjalistycznych, których reprezentacja była nieproporcjonalna w stosunku do faktycznych możliwości (ZSRR — 4 referaty, w tym 3 zamówione, Bułgaria — 2, NRD, Czechosławcja i Węgry — po 1).

Nieco lepiej obecność Polski była widoczna w grupie przewodniczących sesji, w której to roli wystąpiło 4 przedstawicieli naszego kraju (prof. prof. L. Łukaszewicz, A. Straszak i W. M. Turski oraz dr M. Dąbrowski).

Wśród referentów można było spotkać wiele sławnych nazwisk, które praktycznie weszły już do historii informatyki. W delegacji radzieckiej było wielu wybitnych naukowców z profesorami Głuszkowem i Jerszowem na czele.

Całość imprezy trwała pełnych 6 dni, przy czym sesje specjalistyczne odbywały się w godzinach od 9^{00} do 12^{45} oraz od 14^{15} do 17^{30}. Zgodnie ze zwyczajem poprzednich kongresów referaty i dyskusje odbywały się w języku angielskim bez równoczesnego tłumaczenia na inne języki, jak to jest powszechnie praktykowane na innych wielkich konferencjach naukowych. Stąd przewodniczący IFIP prof. H. Zemanek w przemówieniu inauguracyjnym zaapelował do referentów anglosaskich, by pamiętając również o cudzoziemcach, wygłaszali swe referaty wyraźnie i wolno.

Inauguracyjną plenarną sesję Kongresu zaszczycił obecnością król Szwecji Karol Gustaw XVI, który wygłosił krótkie przemówienie okolicznościowe i dokonał formalnego otwarcia. Na sesji tej przemówienia powitalne wygłosili również: przewodniczący sesji, a zarazem Komitetu Organizacyjnego — P. Svenonius, dotychczasowy przewodniczący IFIP — prof. H. Zemanek, Przewodniczący Komitetu Programowego, a zarazem nowy przewodniczący IFIP na okres 1974—77 — prof. H. Freeman, S. Sem-Sandberg, przewodniczący Szwedzkiego Stowarzyszenia Przetwarzania Informacji, a zarazem dotychczasowy wiceprzewodniczący IFIP oraz przedstawiciel UNESCO — prof. Nowożyłow. Uroczystą oprawę muzyczną ceremonii otwarcia zapewniła Królewska Orkiestra Symfoniczna. Końcowym akcentem sesji inauguracyjnej było wystąpienie p. Alvy Myrdal, b. szwedzkiego ministra d/s rozbrojenia, która wygłosiła interesujące przemówienie na temat aktualnych problemów społecznych i politycznych komputeryzacji pt. „Nauka, poufność oraz siła" *(Science, secrecy and power)*. Po wystąpieniu p. Mydral uczestnicy sesji inauguracyjnej zostali uraczeni niezwykłymi efektami dźwiękowymi muzyki komputerowej, opracowanymi przez Studio Muzyki Elektronicznej w Sztokholmie (utwór *"Dator Lotus"* kompozycji Zaida Holmina).

Jeśli chodzi o część roboczą Kongresu, to niewątpliwie najbardziej interesującymi fragmentami były dyskusje panelowe, na które — jak już wspomniano — organizatorzy położyli największy akcent.

Niektóre z tych dyskusji przez samą atrakcyjność problematyki przyciągały rekordową liczbę słuchaczy, zbliżoną do 1000 osób, czego nie można było stwierdzić na niektórych sesjach tematycznych, gdzie wspomniany, słaby poziom merytoryczny lub nadmierne zawężenie treści referatów, połączone najczęściej z całkowitym brakiem umiejętności referowania przez autora, powodowało, że przestronne sale posiedzeń przedstawiały dość żałosny widok.

Sądzę, że interesujące będzie dosłowne przytoczenie tytułów dyskusji panelowych, które odzwierciedlają podstawowe kierunki zainteresowań oraz problemy nurtujące aktualnie środowiska informatyków w krajach o wysokim stopniu rozwoju komputeryzacji. Niektóre z tych problemów pojawiły się w dyskusjach już w naszym kraju, natomiast pozostałe pojawią się niewątpliwie w najbliższym czasie. Tytuły te brzmiały następująco:

1) Programowanie strukturalne *(Structured Programming)*.
2) Komputery w twórczości artystycznej *(Computers in Creative Arts)*
3) Szkolenie w użytkowaniu komputerów do celów zarządzania *(Training in the Use of Computers in Management)*
4) Programowanie w latach osiemdziesiątych *(Programming in the 1980's)*
5) Zarządzanie środkami i możliwościami komputeryzacji *(Manage-*

ment of Computing Resources and Facilities)

6) Poglądy międzynarodowe na systemy informacyjne kierownictwa *(International Views of Management Information Systems)*

7) Zastosowanie komputerów do kształcenia kadr *(Use of Computers in Instruction)*

8) Zdecentralizowane przetwarzanie danych gospodarczych *(Distrubuted Computing for Business Data)*

9) Analiza kosztów i efektów systemów opartych o bazę danych *(Cost-Benefit Analysis of Data-Base Systems)*

10) Projektowanie wspomagane komputerem w przemyśle *(Computer-Aided Design in Industry)*

11) Analiza komputerowa rysunków liniowych *(Computer Analysis of Line Drawings)*

12) Wpływ komputerów na społeczeństwo *(Social Impact of Computers)*

13) Trendy w dziedzinie rozwoju sprzętu i spojrzenie perspektywiczne *(Hardware Trends and Future Outlook)*

14) Problematyka automatów *(Robotics)*

15) Zastosowania komputerów dla potrzeb społeczeństwa *(Computer Applications to Society)*

16) Systemy niewrażliwe na uszkodzenia *(Fault Tolerant Systems)*

17) Oprogramowanie odpłatne: obecny wpływ i przyszłe tendencje *(Software Products: Present Impact and Future Trends)*

18) Zachowywanie tajemnicy i komputery *(Privacy and Computers)*

Zbiór wszystkich wygłoszonych referantów został wydany i przekazany każdemu uczestnikowi Kongresu. Zbiór ten pod nazwą *„Information Precessing 74" (Congress Preprints)* obejmuje 5 oprawnych części formatu A_4 odpowiadających wspomnianemu podziałowi tematycznemu referatów, z tym jednak, że referaty sekcji problemowych 4 (zastosowania naukowe i techniczne) oraz 5 (zastosowania w naukach społecznych i humanistyce) połączono razem w tomie czwartym materiałów. Łączna objętość tych materiałów wynosi 1095 stron. W materiałach tych ze zrozumiałych względów nie ma relacji z przebiegu dyskusji panelowych, które pojawią się dopiero w tradycyjnej ostatecznej wersji materiałów *(Congress Proceedings)*. Końcowa plenarna sesja Kongresu, której przewodniczył prof. Freeman zawierała oprócz samej ceremonii zamknięcia obrad bardzo interesujące 45-minutowe wykłady *(invited lectures)*, a mianowicie:

1) R. M. Davis, dyrektor Instytutu Informatyki w National Bureau Of Standards w Waszyngtonie, pt. „Komputery i międzynarodowa równowaga sił", dotyczący politycznych skutków komputeryzacji

2) W. F. Bodmera, profesora genetyki Uniwersytetu w Oxford pt. „Geny i ludzie", ukazujący nowy, olbrzymi zakres zastosowań komputerów w dziedzinie nauk biologicznych i społecznych,

3) A. Danzina, dyrektora francuskiego Instytutu Informatyki IRIA pt. „Współpraca międzynarodowa w pracach badawczych w dziedzinie informatyki", postulujący wzmożenie tego rodzaju współpracy.

Podsumowując pierwsze ogólne wrażenia z kongresu sztokholmskiego należy stwierdzić, że pomimo braku bardziej sensacyjnych czy przełomowych doniesień naukowych, co charakteryzuje aktualne stadium rozwoju informatyki, impreza ta spełniła swe podstawowe zadanie, jakim jest nawiązywanie bezpośrednich kontaktów oraz wymiana poglądów pomiędzy specjalistami.

Potwierdzeniem tego jest prawie anegdotyczne oświadczenie jednego z członków delegacji polskiej, że dopiero w Sztokholmie dowiedział się w bezpośredniej rozmowie z innym informatykiem polskim o interesujących go pracach prowadzonych w jednym z krajowych ośrodków na terenie innego miasta, o istnieniu których, mimo stosunkowo popularnej tematyki dotychczas nie był poinformowany.

Władysław Klepacz

Rozwój współpracy w Jednolitym Systemie

Na początku maja 1974 r. odbyło się w Warnie kolejne posiedzenie międzyrządowej komisji d/s Jednolitego Systemu. Podjęto liczne uchwały w sprawie dalszego rozwoju prac nad stworzeniem jednolitej bazy sprzętowej dla potrzeb przetwarzania danych w krajach socjalistycznych. Ustalono, też, że we wrześniu br., z okazji 16 Międzynarodowych Targów w Brnie, zorganizowana zostanie ekspozycja znacznej części dotychczasowych osiągnięć technicznych.

Główny Konstruktor NRD d/s Jednolitego Systemu — Dr Manfred Günther — udzielił czasopismu RECHENTECHNIK-DETENVERARBEITUNG wywiadu na temat znaczenia wspomnianego posiedzenia oraz przewidywanych ulepszeń konstrukcyjnych i technologicznych sprzętu. Dr Günther stwierdził, że komisja międzyrządowa pozytywnie oceniła postęp w zakresie dotychczasowych metod współpracy, przeprowadzono m.in. w bieżącym roku pierwsze wspólne badania pakietów programów aplikacyjnych, w ramach wspólnych prac rozwojowych.

Pierwsze badania przeprowadzono z pełnym sukcesem w NRD. W celu zaopatrywania zainteresowanych użytkowników we wspomniane pakiety programów — Centrum Koordynacyjne opracowało wydawnictwo informacyjne, tłumaczone obecnie przez Kombinat ROBOTRON dla użytkowników NRD. Część pierwsza tego wydawnictwa zawiera opis 230 programów aplikacyjnych oraz projektów systemów automatyzacji zarządzania (ASU).

Kraje członkowskie poinformowały ponadto na posiedzeniu w Warnie, że do produkcji seryjnej wdrożono już 95 różnych urządzeń i systemów komputerowych.

Jeszcze w tym roku do produkcji seryjnej wprowadzi się dalszych 39 urządzeń. Wzajemne dostawy sprzętu rozpoczęły się w zasadzie w roku 1973, a główną uwagę konstruktorów urządzeń i systemów skierowano na opracowanie programów testujących oraz obsługi technicznej użytkowników. Jeśli chodzi o działalność narodowych organizacji obsługi użytkowników Sprzętu Jednolitego Systemu, stwierdzono poprawę form współpracy. Tym niemniej konieczne jest dalsze podnoszenie poziomu konserwacji technicznej sprzętu Jednolitego Systemu, co warunkuje efektywność jego eksploatacji.

Na pytanie dotyczące form dalszych prac nad Jednolitym Syste-

mem, dr Günther poinformował, że Komisja Międzyrządowa zaleciła przygotowanie wielostronnego porozumienia w sprawie specjalizacji produkcji w krajach członkowskich. Komisja podjęła również decyzję na temat opracowań konstrukcyjnych w dziedzinie minikomputerów oraz komputerów do sterowania procesami. Między innymi dla kierowania pracami badawczo-rozwojowymi w tej dziedzinie utworzono nową Radę Głównych Konstruktorów. W celu stymulowania wymiany pakietów programów aplikacyjnych zatwierdzono uprzednio przygotowany plan tych prac i ustalono metody i technologię tej wymiany.

Oprócz tego Komisja Międzyrządowa postanowiła utworzyć specjalną grupę roboczą do spraw kierowania i koordynacji wspólnych prac nad stworzeniem zautomatyzowanych systemów projektowania, konstruowania oraz technicznego przygotowania produkcji. Grupa ta w pierwszej fazie działalności zajmie się problemem automatyzacji prac projektowych i konstrukcyjnych w przemyśle maszynowym i w budownictwie.

Na pytanie o prace nad dalszym rozwojem konstrukcji sprzętu produkowanego seryjnie oraz zasadniczych cech charakterystycznych tego rozwoju — dr Günther scharakteryzował wspólny program dalszych ulepszeń Jednolitego Systemu. Prace nad tym programem rozwijają się w następujących kierunkach:

— poprawa stosunku ceny do wydajności sprzętu przez zastosowanie nowych lub udoskonalonych elementów oraz technologii

— dalsze konsekwentne ujednolicanie urządzeń i systemów

— rozszerzanie i uzupełnianie asortymentu urządzeń zewnętrznych, w pierwszym rzędzie pamięci zewnętrznych urządzeń do zdalnego przetwarzania oraz wprowadzania i wyprowadzania danych

— tworzenie nowych wersji systemów operacyjnych uwzględniających większe udogodnienia dla użytkownika.

Na pytanie dotyczące kierunku rozwoju urządzeń do rejestracji i przechowywania danych, dr Günther potwierdził zdarzające się przypadki niedostatecznie efektywnego wykorzystania wysokosprawnych komputerów Jednolitego Systemu wskutek niepełnego asortymentu wydajnych urządzeń tego typu. Podczas narady w Warnie Komisja Międzyrządowa zwróciła uwagę krajów członkowskich na konieczność przyspieszenia rozwoju urządzeń do zdalnego przetwarzania danych, a zwłaszcza stacji abonenckich oraz urządzeń końcowych. Jednocześnie zalecono wypróbowanie systemów zdalnego przetwarzania danych oraz stworzenie eksperymentalnych wzorów takich systemów w oparciu o sprzęt Jednolitego Systemu.

Szczególnie intensywnie prowadzi się również prace nad rozszerzeniem asortymentu pamięci zewnętrznych. W ZSRR oraz w Bułgarii przygotowuje się uruchomienie seryjnej produkcji pamięci dyskowych wymiennych o zwiększonej pojemności. Aktualnie w Bułgarii przygotowuje się konstrukcję pamięci dyskowej o pojemności 30 Mbajtów.

W ramach dalszego rozwoju Jednolitego Systemu podjęto również prace w kierunku innych rodzajów urządzeń zewnętrznych, jak np. systemu wprowadzania, przechowywania i wyprowadzania informacji na mikrofilmie z możliwością współpracy z komputerami. Jednocześnie realizuje się wielostronną współpracę w dziedzinie stworzenia całego systemu alfanumerycznych monitorów ekranowych.

Na ostatnie pytanie dotyczące wystawy Jednolitego Systemu przy okazji 16 Międzynarodowych Targów w Brnie dr Günther zapowiedział, że centralnym punktem ekspozycji NRD będzie komputer EC 1040. Odbędzie się też sympozjum obejmujące referaty zarówno o komputerze, jak i o zebranych w NRD doświadczeniach z jego eksploatacji oraz eksploatacji pakietów programów aplikacyjnych.

Komisja Międzyrządowa poleciła utworzenie specjalnej grupy roboczej d/s międzynarodowych Wystaw i Targów. Grupa ta czuwać będzie nad tym, aby podczas przyszłych wystaw na pierwszy plan ekspozycji wysuwać przykłady możliwości zastosowań wystawianego sprzętu. Taki właśnie akcent dało się już odczuć w czasie wystawy w Brnie.

W.K.

Międzynarodowe kursy EPD w Budapeszcie

Nauczanie maszynowego przetwarzania danych w formie kursów zapoczatkowano na Węgrzech już w 1953 w Centralnym Urzędzie Statystycznym.

Celem dalszego rozwoju tego rodzaju działalności utworzono w 1969 r. Węgierskie Centrum Szkoleniowe Przetwarzania Danych (SZAMOK), które od tej chwili stało się podstawową instytucją szkolenia informatyków.

Mając stałą siedzibę w Budapeszcie oraz sieć oddziałów na terenie kraju SZAMOK troszczy się o kształcenie analityków systemów, programistów, programistów systemowych, operatorów i konserwatorów w trybie szkolenia podyplomowego.

W interesie dokształcania już wyszkolonych specjalistów informatyków i kierowników instytucji gospodarczych SZAMOK regularnie organizuje każdej jesieni i wiosny tzw. „kursy specjalne i uzupełniające wiedzę". Na podstawie porozumienia pomiędzy rządem węgierskim a Programem Rozwoju Narodów Zjednoczonych SZAMOK od stycznia 1973 kontynuuje swą działalność jako instytut międzynarodowy, co zostało podkreślone w jego nowej nazwie Międzynarodowe Centrum Szkoleniowe Przetwarzania Danych.

W konsekwencji tego faktu SZAMOK organizuje kursy nie tylko dla obywateli węgierskich, lecz również cudzoziemców, w pierwszym rzędzie z krajów rozwijających się. Kursy te prowadzone są w językach rosyjskim i angielskim. Wykłady na kursach prowadzone są częściowo przez własny personel dydaktyczny o wyższych kwalifikacjach zawodowych i znajomości języków obcych, a częściowo przez ekspertów zagranicznych, którzy wykonują swe funkcje dydaktyczne na statusie zaproszonych wykładowców.

SZAMOK przyczynia się również do rozpowszechniania zastosowań produkowanych w krajach RWPG komputerów Jednolitego Systemu dzięki nauczaniu na kursach programowania i konserwacji technicznej tych maszyn.

SZAMOK zawarło również umowy z producentami sprzętu komputerowego. W oparciu o te umowy Centrum kształci przyszłych specjalistów dla użytkowników, w szczególności programistów i operatorów.

Pracę kursantów ułatwiają w dalszym stopniu wydawane przez SZAMOK publikacje. Również zatrudnieni w działalności praktycznej specjaliści mogą wiele skorzystać z literatury fachowej Centrum. Kompleks budynków Centrum zostanie oddany do użytku w 1975 roku. Kompleks ten, zawierający nowocześnie wyposażone pomieszczenia dydaktyczne, sieć telewizji przemysłowej oraz bibliotekę fachową, pozwoli podnieść jakość szkolenia, natomiast hotel i restauracja zapewnią słuchaczom z kraju i zagranicy odpowiednie warunki zakwaterowania i wyżywienia.

W nowej siedzibie Centrum zostanie wyposażone również w nowoczesny komputer. Przyczyni się on nie tylko do wzrostu efektywności procesu dydaktycznego, ale również stworzy słuchaczom możliwość praktycznego sprawdzenia wiedzy zdobytej w czasie wykładów teoretycznych.

Praca ODRY 1305 w systemie operacyjnym GEORGE-2

W bieżącym roku w Resortowym Ośrodku Informatyki i Doradztwa Organizacyjnego ETOCHEM zainstalowano i uruchomiono elektroniczną maszynę cyfrową ODRA-1305 wyposażoną w następujące urządzenia zewnętrzne: trzy jednostki dysków wymiennych, osiem jednostek taśm magnetycznych, czytnik kart 80-kolumnowych, czytnik taśmy papierowej, perforator taśmy papierowej, drukarka wierszowa.

Fakt zainstalowania dysków, bardzo szybkiego czytnika kart (2000 kart/min.) i najszybszych jednostek taśm magnetycznych a także cykl podstawowy jednostki centralnej wynoszący 1 mikrosekundę, zalicza ten komputer do rzędu szybkich w stosunku do techniki światowej. Dzięki tym zaletom ODRA-1305 stawia bardzo wysokie wymagania swoim użytkownikom. Aby w należyty sposób wykorzystać jej szybkość należy sięgnąć do bardziej nowoczesnych metod pracy.

Wszystkie współczesne komputery do przetwarzania danych pracują wyłącznie w systemach operacyjnych. Pierwszym systemem operacyjnym dla maszyn serii ODRA-1300 porównywalnym z opracowaniami takich firm, jak IBM i CDC jest GEORGE 3 lub GEORGE 4, który jednak wymaga minimalnej pamięci operacyjnej 64 Ksłów. W obecnej chwili nie ma jeszcze technicznej możliwości pracy maszyn ODRA-1305, w tym systemie, co zresztą ma nastąpić — według zapowiedzi ELWRO — na przełomie lat 1974/75.

Jednak już obecnie istnieje możliwość pracy w uproszczonym systemie operacyjnym GEORGE 2. System ten — w przypadku maszyn wyposażonych w pamięć na dyskach magnetycznych — pozwala na uzyskanie automatycznej wieloprogramowości i — zgodnie z opinią specjalistów angielskich — daje oszczędność 30—60% czasu komputera w zależności od rodzaju pracy.

W Resortowym Ośrodku Informatyki i Doradztwa Organizacyjnego ETOCHEM już w początkowym okresie uruchomienia komputera stwierdzono brak dostatecznej ilości czasu komputera. Stan ten wywołany został głównie przez dwa czynniki. Pierwszym z nich było prowadzenie eksploatacji przez komunikowanie się operatora z komputerem za pośrednictwem monitora, drugim — brak ugruntowanych doświadczeń organizacyjnych. Tak więc w celu zwolnienia ukrytych rezerw czasu maszyny wprowadzenie systemu operacyjnego stało się koniecznością już w pierwszym okresie, czego w zasadzie nie przewidziano przy planowaniu uruchamiania ośrodka.

1 lipca br. dokonano praktycznego wdrożenia systemu operacyjnego GEORGE 2, stwierdzając jednocześnie jego całkowitą niezawodność. Pomimo dużej awaryjności komputera w pierwszym okresie eksploatacji i powstałej — przy przejściu na wyższy szczebel technologiczny przetwarzania — bariery ludzkiej, okazało się, że przydział czasu na prace objęte systemem operacyjnym można było zmniejszyć o 2/3, przy czym jednocześnie zniknął problem niezaspokajania zgłaszanych potrzeb zespołów projektowo-programowych.

W pierwszym etapie w zakres prac objętych systemem operacyjnym włączono tłumaczenie (kompilację) programów napisanych w językach PLAN i COBOL, a także ich wstępne uruchomienia. Do tłumaczeń programów użyto kompilatorów współpracujących z dyskami, jako podstawowymi urządzeniami zewnętrznymi. Zaletą ich jest kilkakrotne skrócenie czasu pracy w porównaniu z techniką opartą na taśmach magnetycznych, wadą — niemożność pracy seryjnej z powodu konieczności użycia tzw. konsolidatora po każdorazowym pojedynczym tłumaczeniu programu scalającego skomplikowany program własny.

Przyjęto więc następującą metodę działania. W pierwszej fazie programy źródłowe są w sposób seryjny zapisywane w zbiorze programów źródłowych na taśmie magnetycznej. Istnieje też możliwość dokonywania zmian (wprowadzanie poprawek, ulepszeń) wewnątrz programów zapisanych w tym zbiorze w jednym z poprzednich przebiegów, a także — kasowania programów zdezaktualizowanych. Faza druga obejmuje szereg pojedynczych tłumaczeń wybranych programów z tego zbioru i zapisywania ich w postaci gotowej do pracy w specjalnie do tego celu przeznaczonym zbiorze dyskowym. Ponieważ przy tłumaczeniu programu napisanego w języku COBOL muszą brać udział dwa programy standardowe (tj. kompilator i konsolidator), a przy tłumaczeniu programu napisanego w PLANie aż trzy — faza ta wymaga wykonania całego szeregu czynności operatorskich. Wykonywanie ich w systemie tradycyjnym, tj. ręcznie wydłużyłoby czas pracy maszyny 2—3-krotnie, natomiast w systemie operacyjnym wszystkie te czynności wykonywane są automatycznie bez interwencji operatora.

Do obowiązku programisty należy przygotowanie programów źródłowych lub poprawek do nich i przekazanie ich służbie kompletacyjnej, która układa je w pożądanej kolejności i tak przygotowaną pracę oddaje do wykonania służbie obsługi komputera.

Taka metoda pracy pozwoliła na osiągnięcie jeszcze jednej dodatkowej korzyści. Do czasu wprowadzenia systemu operacyjnego i zblokowanej kompilacji każdy zespół projektowo-programowy posiadał własne zbiory programów źródłowych i programów przetłumaczonych (tzw. binarnych). Utworzenie jednego, wspólnego dla całego ośrodka zbioru umożliwiło przekazanie sporej liczby taśm magnetycznych do innych celów.

Także i uruchamianie programów odbywa się w podobny sposób. Programista jest zobowiązany oddać przygotowaną pracę do zespołu kompletacyjnego, a ten przekazuje ją do realizacji obsłudze komputera.

Innym ważnym osiągnięciem systemu operacyjnego jest fakt, że źle przygotowana praca powoduje wprawdzie odsunięcie terminu uzyskania ostatecznych wyników przetwarzania (co zresztą następuje i w tradycyjnym systemie pracy), praktycznie jednak nie powoduje zużycia czasu komputera. Ponadto uzyskuje się automatyczne rejestrowanie i rozliczenie czasu prac prowadzonych w systemie operacyjnym. Otrzymanie takiego rozliczenia jest niemożliwe w systemie ręcznym. Rozliczenie w systemie operacyjnym nosi także charakter analityczny w stosunku do jakości pracy programisty podając informację o popełnionych przez niego błędach w przygotowaniu pracy i niweluje konflikty między służbą operatorską a programistą dostarczając obiektywnych danych na drukowanych dokumentach.

W rozliczeniu oszczędności czasu zawarto także pewne wahania w sprawnej eksploatacji wywołane ujawnianiem nieuświadamianych poprzednio niedociągnięć organizacyjnych. Dozorowanie przebiegu w systemie operacyjnym na sali komputera wymaga znacznie niższych umiejętności operatora maszyny — operuje on w zasadzie tylko programami systemu operacyjnego, co nie jest trudne. Wymagane są natomiast znacznie wyższe kwalifikacje programistów, a także zespołu kompletującego prace na EMC i eksploatującego systemy.

W zamierzeniach ETOCHEM leży opracowywanie projektów technicznych systemów z włączeniem oprogramowania pracy komputera w systemie operacyjnym. Wystarczy bowiem uświadomić sobie, że możliwość uzyskania przeciętnie 50% oszczędności czasu pracy komputera jest równoważne wybudowaniu nowego ośrodka wyposażonego w podobną maszynę.

Krzysztof Marks
ETOCHEM

od str. 47 do str. 50

System zarządzania bazą danych SYKON

Dla ilustracji problemów omawianych w tym cyklu artykułów przedstawiamy jeden ze zrealizowanych przykładów systemów zarządzania bazą danych — SYKON. SYKON (System Konwersacyjny) pomyślany był jako oprogramowanie części Systemu Informacji Kierownictwa dla Krajowego Biura Informatyki, a mianowicie banku danych z dziedziny informatyki (BDI). Z uwagi na rozległość dziedziny zainteresowań KBI jak też i niemożliwość zdefiniowania w formie zrutynizowanych procedur funkcji spełnianych przez tę instytucję, jedynym możliwym było rozwiązanie oparte o ideę generalizowanych systemów zarządzania bazą danych.

System Zarządzania Bazą Danych SYKON (SZBD SYKON), należy do klasy tzw. systemów samowystarczalnych *(self contained)* i przeznaczony jest do obsługi sformatowanych baz danych. Jest on narzędziem umożliwiającym zaspokojenie potrzeb informacyjnych użytkownika poprzez obsługę funkcji dostępu, wyszukiwania, przetwarzania i aktualizacji informacji zawartych w bazie (bazach) danych. Proces wyszukiwania, przetwarzania i aktualizacji może odbywać się zarówno w trybie przetwarzania konwersacyjnego, jak i przetwarzania wsadowego (partiowego). Odpowiednio do wymienionych trybów przetwarzania SZBD SYKON dzieli się na dwie części:

- Moduł obsługi przetwarzania wsadowego
- Moduł obsługi dostępu, wyszukiwania i aktualizacji bazy danych w trybie zdalnego przetwarzania na bieżąco.

SYKON narzuca użytkownikowi własną fizyczną strukturę danych, zapewniającą dostatecznie dużą elastyczność logicznej struktury danych i szybką realizację procesu wyszukiwania. Organizacja bazy danych z punktu widzenia procesu wyszukiwania oparta jest o znaną metodę zbiorów odwróconych. Elanych zapewnia aparat dynamicznego budowania tych struktur, zawarty w języku SYKON (instrukcja LINK) oraz — nie „przeszkadzający" temu procesowi — odpowiedni opis logiczny bazy danych, który nie zawiera, a tym samym i nie ustala, logicznych powiązań w bazie danych.

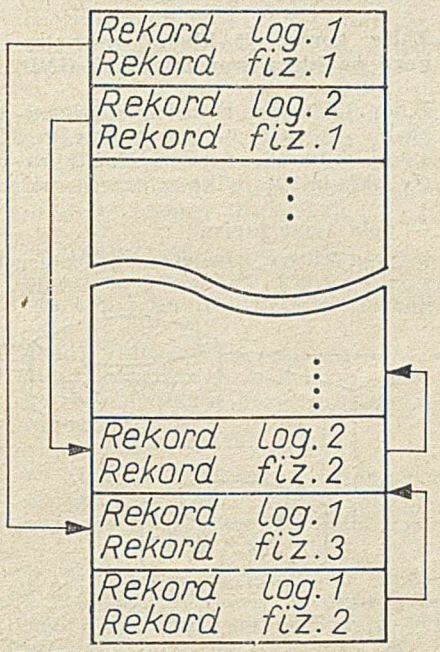
Rozwiązanie to jest przeciwieństwem rozwiązań proponowanych przez DBTG [1] komitetu CODASYL, co nie było bez znaczenia przy wyborze SYKON-u jako przedstawianego tu przykładu SZBD.

[1] Data Base Task Group, Raporty z lat 1969, 1971 i 1973.

Organizacja bazy danych

SZBD SYKON może jednocześnie obsługiwać dowolną liczbę baz danych. Wszystkie one mają jednorodną fizyczną organizację. Baza danych systemu SYKON składa się z następujących zbiorów [2]

- zbioru zawierającego opis bazy danych (xxxSLD) [3], zwany też słownikiem danych
- zbioru podstawowego (xxxINF), jedynego zbioru, który faktycznie zawiera informacje użytkownika
- dwóch zbiorów stanowiących łącznie zbiór odwrócony zbioru xxxINF(xxxWDS) i (xxxSEL)
- zbiór konwersji kluczy zewnętrznych zapisów (rekordów) na wewnętrzne klucze bazy danych (xxxIND).



Rys. 1

[2] Słowo zbiór używane jest w takim sensie jak „widzi" go system operacyjnych *(file)*.

[3] Sześcioznakowe nazwy zbiorów identyfikują bazę danych (xxx-dowolne trzy znaki rozpoczynające się od litery) i zbiór w ramach bazy danych (SLD, INF, WDS, SEL i IND).

Zbiór podstawowy (INF)

Zbiór podstawowy bazy danych jest zbiorem o dostępie bezpośrednim. Ma on zmienną i teoretycznie nieograniczoną długość zapisów logicznych, które są identyfikowane i adresowane przez klucz zwany kluczem wewnętrznym (KLWx). Klucz wewnętrzny nadawany jest każdemu zapisowi logicznemu jednorazowo w momencie jego otwierania. Klucz wewnętrzny ma format 24-bitowej liczby binarnej bez znaku, co zapewnia możliwość zapisania w bazie danych maksymalnie 16.777.215 zapisów. Nowe są dopisywane bezpośrednio za ostatnim zapisem w obszarze pierwotnym zbioru (patrz rys. 1). Każdy zapis składa się z jednego podzapisu kluczowego i dowolnej liczby podzapisów innych typów, podzapis zawierać może dowolną liczbę danych. W roli identyfikatora podzapisu kluczowego występuje klucz zewnętrzny zapisu (KLZx), natomiast każdy z podzapisów musi zawierać unikalny w ramach zapisu własny klucz podzapisu (KPxx). W zapisie kluczowym zapisywane są obok dowolnych informacji użytkownika następujące informacje systemowe:

- klucz wewnętrzny (KLWx)
- typ zapisu (TYPx)
- tablice podzapisów (TPRx).

Dane użytkownika, ich adresy względem początku podzapisów, formaty, długości i inne niezbędne atrybuty danych są zdefiniowane przez słownik danych, który stanowi oddzielny zbiór zwany xxxSLD.

Każdy zapis logiczny składa się z dowolnej liczby zapisów fizycznych powiązanych łącznikami adresowymi (patrz rys. 2). Długość zapisu fizycznego jest stała i ustalana przez użytkownika.

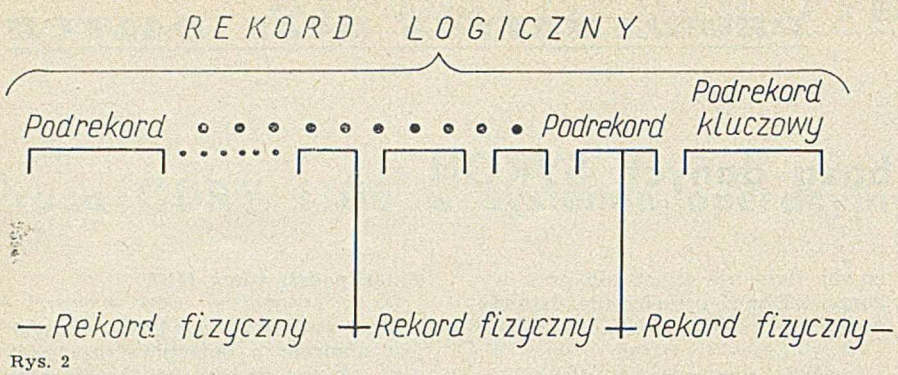
Słownik danych (SLD)

Słownik danych jest zbiorem, w którym zawarty jest opis formalny wszystkich gromadzonych w zbiorze podstawowych danych.

Zbiór składa się z zapisów o stałej długości. Każdy zapis zawiera opis jednej danej. Opis danej zawiera między innymi:

- symbol danej — unikalny w ramach zbioru czteroznakowy identyfikator danej
- typ podzapisu — dwuznakowy identyfikator; przyjęty jest stały znak do identyfikacji podzapisu kluczowego — „K"

Symbol agregatu | symbol rekordu



Rys. 2

- format danej — kod formatu danej
- długość danej — liczba określająca długość danej w bajtach
- adres danej — liczba określająca adres pierwszego pola przeznaczonego dla danej względem początku podzapisu
- wskaźnik deskryptora — kod przyjmujący wartości Ø, 1, .., 9 określający czy dana jest deskryptorem i ewentualnie przedział skalowania.
- łącznik adresowy do zbioru WDS — adres względny zapisu zawierającego wartości tej danej w zbiorze odwróconym.

Pierwszy zapis zbioru SLD ma różną od wszystkich następnych zapisów budowę i zawiera takie informacje, jak:

— liczba zapisów w zbiorze SLD
— liczba deskryptorów
— długość zapisu fizycznego zbioru podstawowego INF
— maksymalna liczba zapisów fizycznych, z jakiej może się składać jeden zapis logiczny itp.

Słownik danych jest sporządzany przez użytkownika przed założeniem bazy danych.

Zaletą SYKON-u jest możliwość wprowadzania korekt do raz zdefiniowanej bazy danych. W tym celu wystarczy zmienić opis bazy danych (słownik danych).

Zbiór wartości deskryptorów (WDS)

Zbiór WDS jest zbiorem o bezpośrednim dostępie i zmiennej długości zapisów logicznych. Każdy zapis logiczny zbioru WDS dotyczy jednej danej będącej deskryptorem.

Obok każdej wartości danej zapisywany jest łącznik stanowiący adres do odpowiedniego zapisu w zbiorze SEL zawierającego listę adresów kierujących z kolei do odpowiednich zapisów zbioru podstawowego (patrz rys. 3).

Dostęp do zbioru WDS realizowany jest za pośrednictwem łączników adresowych utrzymywanych w zbiorze SLD.

Zbiór łączników SEL

Zbiór SEL jest zbiorem o organizacji z bezpośrednim dostępem i zmiennej długości zapisów. Sposób tworzenia łańcucha zapisów fizycznych i zapisywania do obszaru zbioru jest identyczny jak w przypadku zbiorów INF i WDS.

Jeden zapis logiczny zbioru SEL zawiera adresy względne wszystkich zapisów w zbiorze INF, w których określona dana-deskryptor przyjmuje wartość korespondującą z odpowiednią wartością w zbiorze WDS. Odpowiednią — to znaczy konkretnie z tą, przy której łącznik zawiera adres do omawianego zapisu SEL (patrz rys. 3).

Adresy względne do zbioru INF są to 24-bitowe liczby binarne bez znaku. Zapis logiczny może zawierać dowolną liczbę adresów do zbioru INF.

Zbiór konwersji klucza zewnętrznego na klucz wewnętrznych (IND)

Zbiór IND jest zbiorem o organizacji indeksowo-sekwencyjnej o stałej długości zapisów. Każdy zapis składa się tylko z trzech pól:

— pola typu zapisu
— pola klucza zewnętrznego zapisu
— pola klucza wewnętrznego zapisu.

Zbiór IND służy do nadawania nowo otwieranym zapisom wartości klucza wewnętrznego bazy danych, a ponadto jest wykorzystywany każdorazowo, gdy wymagany jest dostęp do zapisu w zbiorze INF według wartości klucza zewnętrznego.

Powiązania zbiorów danych

Wzajemne powiązania występujące między zbiorami bazy danych przedstawione są na rys. 3.

Zbiór SLD definiuje budowę zapisów w zbiorach INF, IND i WDS zapisy zbioru WDS zawierają powiązanie z zapisami zbioru SEL.

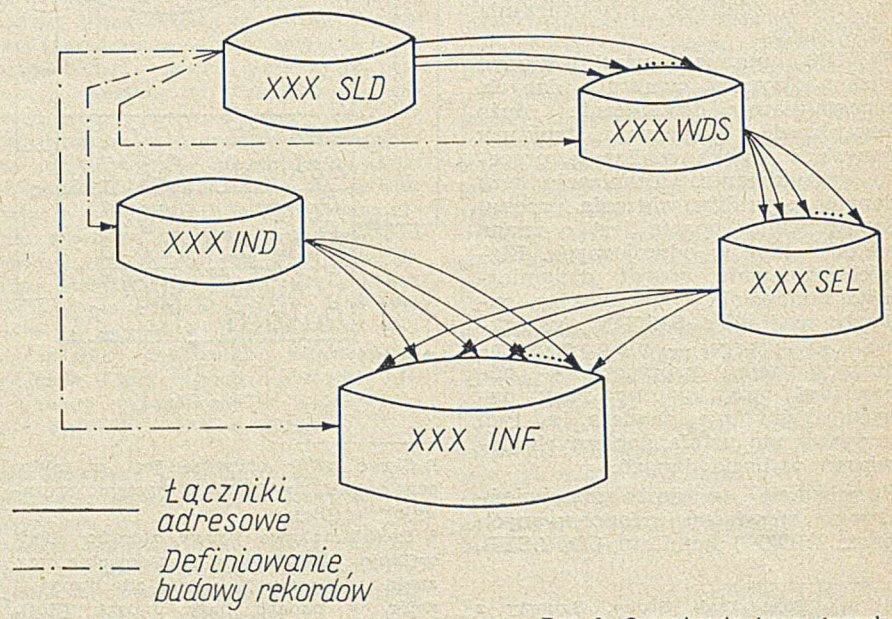
Oba te zbiory tworzą jedną logiczną całość i łącznie są niczym innym, jak zbiorem odwróconym zbioru INF ze względu na dane-deskryptory. Zbiór IND jest również zbiorem odwróconym zbioru INF ze względu na klucz zewnętrzny (KLZx).

Architektura systemu zarządzania bazą danych SYKON

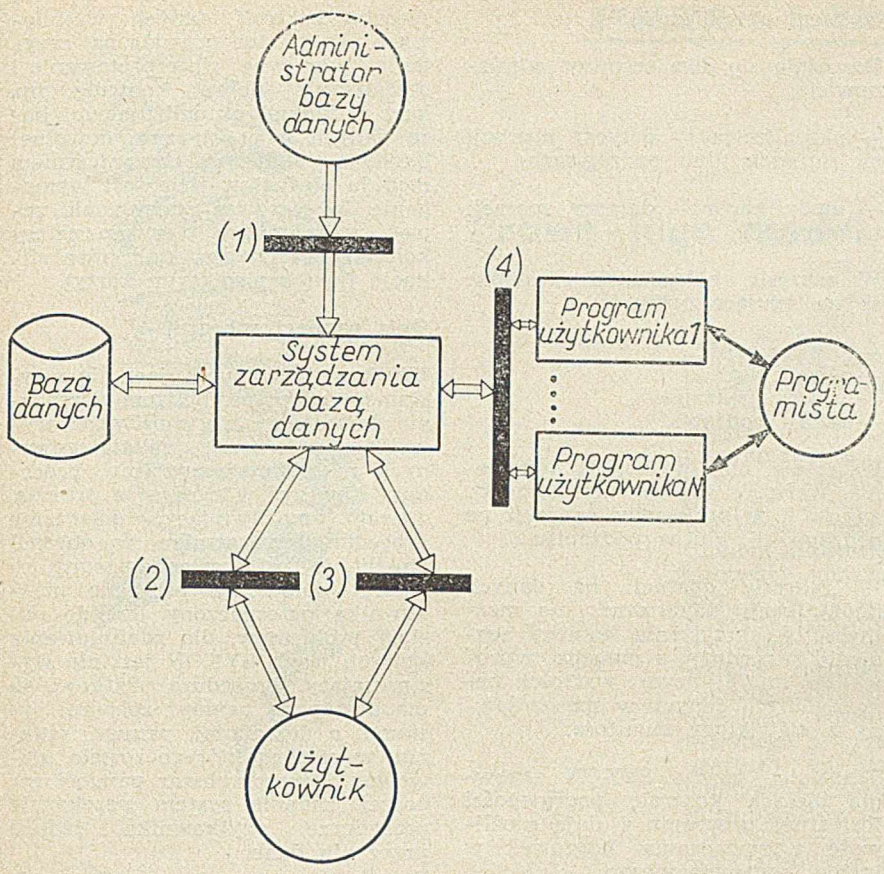
Architektura SZBD SYKON przedstawiona została na rys. 4. Oznaczono tam cztery dostępne w SYKON-ie mechanizmy pośredniczące (interfejsy) między systemem zarządzania bazą danych a jego użytkownikami.

Są to:

- opis bazy danych
- język konwersacyjny SYKON
- język parametryczny SYKON
- opis pamięci roboczej.



Rys. 3. Organizacja bazy danych



Rys. 4. Architektura systemu SYKON

Opis bazy danych

Administrator bazy danych definiuje poprzez opis danych logiczną i, w dopuszczalnym zakresie, fizyczną organizację bazy danych. Na poziomie logicznym definiuje on wszystkie występujące w zbiorze podstawowym typy zapisów i podzapisów poprzez opis wszystkich występujących w nich danych. Opis każdej danej zawiera takie elementy, jak: symbol danej, jej format, adres względem początku zapisu (podzapisu), poziom zaufania itp. Dowolnej danej można nadać w opisie atrybut deskryptora, co powoduje, że dana ta będzie elementem zbiorów odwróconych. Tym samym zapewniona będzie możliwość szybkiego wyszukiwania wg tej danej jako kryterium selekcji. Liczba deskryptorów jest dowolna.

Forma zewnętrzna opisu bazy danych ma prostą postać tabelaryczną.

Język konwersacyjny SYKON

Język konwersacyjny SYKON jest narzędziem przeznaczonym dla użytkownika komunikującego się z systemem za pośrednictwem urządzeń końcowych. Przytoczony poniżej opis języka jest niepełny i nieformalny.

Zadania języka dzielą się na dyrektywy i komendy. Ogranicznikiem zdania jest średnik (;).

Zadanie typu dyrektywa rozpoczyna się znakiem „#". W języku SYKON dostępne są następujące dyrektywy (podkreślone są znaki sprawdzane przez translator):

PROGRAM — identyfikacja programu; określenie typu pracy (dostęp, aktualizacja itp.) oraz określenie urządzenia wyjściowego (monitor ekranowy, drukarka);

SELEKCJA — rozpoczyna segment programu, w którym użytkownik definiuje kryteria selekcji;

PROCEDURA — rozpoczyna segment programu, w którym użytkownik definiuje część proceduralną programu;

AKTUALIZACJA — rozpoczyna segment programu aktualizacji bazy danych (dyrektywa opcjonalna);

WYKONAJ — powoduje wykonanie poprzedzającej tę dyrektywę sekwencji programu poczynając od poprzedniej dyrektywy WYKONAJ lub od początku programu (dyrektywa opcjonalna);

STOP — powoduje zakończenie zadania (dyrektywa opcjonalna).

W poszczególnych segmentach programu, tj. w segmencie selekcji, segmencie proceduralnym i segmencie aktualizacji dostępne są odpowiednie komendy, które są pokrótce poniżej scharakteryzowane.

Segment selekcji

❺ Komenda selekcji
Ogólna postać:

Nazwa danej-deskryptora operator relacji wartość;

gdzie:

nazwa danej-deskryptora jest identyfikatorem danej zdefiniowanej w opisie bazy danych jako deskryptor, operator relacji — w chwili obecnej dostępne są operatory, a mianowicie: >, <, = — większe, mniejsze, równe, wartość — liczba dziesiętna lub dowolny ciąg znaków.

W segmencie selekcji użytkownik może pisać dowolną liczbę komend selekcji, przy czym wszystkie one traktowane są jak gdyby były połączone operatorem iloczynu logicznego. Użycie dyrektywy # WYKONAJ w segmencie selekcji powoduje wykonanie poprzedzającej sekwencji komend selekcji, w wyniku czego dostaje się w odpowiedzi liczbę istniejących w bazie danych takich zapisów, które spełniają zadane warunki.

❻ Komenda dobierania zapisów logicznie powiązanych.
Ogólna postać:

LINK nazwa danej — 1 (typ zapisu-1) = nazwa danej-deskryptora-2 (typ zapisu-2);

Komenda LINK powoduje, że do każdego zapisu typu-1 spełniającego kryteria selekcji — określone na wyższym poziomie selekcji (dobierania) — dołączane są zapisy typu-2 takie, że wartość danej o nazwie danej-1 zapisu nadrzędnego równa jest wartości danej o nazwie-danej-2 z zapisu(ów) dobieranych.

Komenda LINK języka SYKON jest prostym a jednocześnie niezwykle „mocnym" i funkcjonalnym aparatem dynamicznego budowania dowolnych struktur logicznych włącznie ze strukturami sieciowymi zawierającymi cykle.

Idea stworzenia aparatu pozwalającego na dynamiczną budowę struktur logicznych danych, dającego możliwość traktowania bazy danych w sposób różny, w zależności od potrzeb konkretnego aktualnego problemu, jest zdaniem autorów istotnym novum w praktyce projektowania systemów zarządzania bazą danych.

Liczba poziomów selekcji (dobierania) jest w aktualnej wersji translatora ograniczona do 10.

Segment proceduralny

❼ Komenda definiowania raportu wyjściowego (odpowiedzi).
Postać ogólna:

RAPORT = nazwa danej-1, nazwa danej-2, ..., nazwa danej-n;

W liście danych komendy RAPORT użytkownik określa, co ma się pojawić w raporcie wynikowym.

Mogą znaleźć się tam nazwy danych ze słownika bazy danych — ewentualnie z dodatkową ich kwalifikacją (np. z wartością klucza zapisu lub jego dowolną częścią), oraz nazwy zmiennych użytkownika (rozpoczynające się od litery X). Zmienne użytkownika muszą być w programie zdefiniowane poprzez formuły.

● Komenda typu formuła.
Postać ogólna:

$$\text{zmienna użytkownika} = \left\{\begin{array}{l}\text{nazwa danej}\\ \text{liczba}\\ \$\ \text{nazwa danej}\end{array}\right\}$$

$$\left[\text{operator}\ \left\{\begin{array}{l}\text{nazwa danej}\\ \text{liczba}\\ \$\ \text{nazwa danej}\end{array}\right\}\ \ldots\ \right];$$

gdzie:

$ nazwa danej — suma (Σ) wartości tej danej we wszystkich wyselekcjonowanych zapisach.
operator — + (dodaj), — (odejmij), * (pomnóż), / (podziel), ... — (bardzo nieformalnie) oznacza, że formuła może być dalej kontynuowana,

$\left\{\begin{array}{l}a\\ b\\ c\end{array}\right\}$ — może wystąpić jeden i tylko jeden spośród elementów *a*, *b*, *c*,
[] — opcja.

W formułach dopuszczalne jest używanie nawiasów zgodnie z naturalnymi zasadami ich używania.

● Komenda uporządkowania
Postać ogólna:

$$\text{SORT} = K_1, {}_1\emptyset, \ldots K_5, {}_1\emptyset;$$

gdzie:
K_i — klucz sortowania, rolę którego może spełniać dowolna dana z listy komendy RAPORT
Ø — uporządkowanie malejące
1 — uporządkowanie rosnące.

Segment aktualizacji

Rozróżnia się dwa terminy, a mianowicie:

— aktualizacja — dotyczy operacji na zapisach i/lub podzapisach

— modyfikacja — dotyczy operacji na danych.

W zakresie aktualizacji dostępne są następujące operacje:

— otwórz zapis
— skreśl zapis
— otwórz podzapis
— skreśl podzapis.

Wszystkie one wymagają określenia wartości klucza zewnętrznego zapisu i, w przypadku operacji na podzapisach, klucza podzapisu.

W zakresie operacji na danych (modyfikacji) użytkownik ma możliwość wykorzystania komend segmentu procedury, a ponadto, wprowadzać może nowe wartości poszczególnych danych w sposób jawny z klawiatury monitora.

Translator języka SYKON zapewnia bieżącą kontrolę poprawności formalnej programu i daje możliwość wprowadzania poprawek w trybie konwersacyjnym.

Język parametryczny SYKON

Język parametryczny SYKON jest odpowiednikiem języka konwersacyjnego SYKON dla użytkownika komunikującego się z systemem w trybie przetwarzania wsadowego.
W porównaniu z językiem konwersacyjnym nie posiada on aktualnie możliwości, jakie daje komenda LINK, no i z przyczyn oczywistych zupełnie inny jest jego aparat aktualizacji bazy danych. Posiada natomiast dodatkowo możliwości opisu budowy zapisów danych wejściowych (na kartach perforowanych, taśmie magnetycznej) oraz sposobu kontroli danych wejściowych. Aparat ten pozwala na wczytanie, poprawne zinterpretowanie i dostatecznie dobrą kontrolę np. kart z danymi aktualizującymi bazę danych o praktycznie dowolnej budowie. Jedynym ograniczeniem modułu obsługi wejścia jest wymaganie, by pierwsze cztery znaki zapisu wejściowego (pierwsze cztery kolumny karty) zawierały identyfikacje typu zapisu (typu karty).

Opis pamięci roboczej

Użytkownik komunikujący się z systemem za pośrednictwem urządzenia końcowego ma możliwość wywołania własnych, zakatalogowanych w bibliotece systemu, procedur. Aby być w zgodzie z prawdą, aparat umożliwiający dołączanie procedur użytkownika w obecnej chwili nie jest jeszcze dostępny.
Aparat dołączania procedur użytkownika przeznaczony jest do obsługi problemów, dla zdefiniowania których język SYKON jest nie wystarczający. Procedura użytkownika ma dostęp do pamięci roboczej systemu, a dokładniej, pamięci zadania, w ramach którego została wywołana. Poprzez obszar pamięci roboczej zadania system przekazuje procedurze użytkownika żądane przez nią dane.

Pamięć robocza jest stronicowana. System przekazuje procedurze użytkownika adres pamięci roboczej oraz adres formalnego opisu organizacji danych w tej pamięci. Opis pamięci roboczej pozwala na poprawną i łatwą jej interpretację przez procedurę użytkownika. Procedury użytkownika mogą być pisane w językach ASSEMBLER i PL/1.

Do systemu SYKON wracać będziemy jesezcze niejednokrotnie przy okazji omawiania poszczególnych problemów związanych z Systemami Zarządzania Bazą Danych.

Władysław Bogucki
Witold Staniszkis

SPROSTOWANIE

Biuro Techniczne firmy CONTROL DATA w Warszawie przeprasza wszystkich zainteresowanych za nieścisłość w treści zamieszczonej w nr 6/74 informacji na temat możliwości zakupu dwóch kontrolnych pakietów dyskowych CDC, ponieważ wszelkie tego typu transakcje w PRL mogą być przeprowadzane **wyłącznie za pośrednictwem polskich central handlu zagranicznego.**

Prawo - jego model cybernetyczny

Recenzowana praca ukazała się w niewielkim nakładzie przed czterema laty. Już wkrótce po wydaniu stała się niedostępna na rynku księgarskim, mimo iż w tym czasie cybernetyka zaledwie torowała sobie drogę wśród teoretyków prawa, nie docierając swym wpływem do właściwego środowiska prawniczego, tzn. do rzeszy praktyków prawa. Obecnie zachodnioniemieckie wydawnictwo J. Schweitzer Verlag przygotowuje do druku tłumaczenie tej pracy dla niemiecko-językcznych krajów Europy oraz dla czytelników amerykańskich.

Powodem zainteresowania pracą polskiego autora jest pionerski charakter. Dotychczasowe publikacje książkowe na temat zastosowań cybernetyki do prawa miały postać przyczynków lub omawiały cybernetykę jako aparat metodologiczny [1]. Recenzowana praca jest w literaturze europejskiej — a o ile mi wiadomo, także w światowej — pierwszą udaną próbą ujęcia bardzo istotnych problemów państwa i prawa w języku cybernetyki i matematyki. Dojrzałe wywody autora są owocem niespełna trzydziestoletniej pracy teoretycznej i bogatej, równoległej praktyki prawniczej. Stąd znaczenie tej książki, ukaże zapewne dopiero przyszłość, a na razie jej wartość może być bądź niedostrzegana, bądź traktowana jako zagrożenie własnego dorobku teoretycznego, w historii nauki fenomen bynajmniej nie odosobniony.

Teoria A. Kiszy jest przykładem pozytywnej odpowiedzi na rozlegające się ostatnio nazbyt często, zarówno wśród uczonych polskich, jak i radzieckich, głosy krytyczne, z jednej strony — sygnalizujące pozostawienie nauk społecznych w tyle za rozwojem nauk ścisłych, z drugiej — nawołujące do oparcia nauki prawa na podstawach metodologicznych statystyczno-matematycznych [2].

Paradoksem jest, że przy znanym nienadążaniu teorii programowania za rozwojem maszyn cyfrowych, co skłania bardziej radykalnych autorów do twierdzenia, że taka teoria jeszcze nie istnieje [3], informatyka prawnicza, będąca najsłabiej rozwijającym się odcinkiem informatyki i oparta, jak dotychczas, na prymitywnej koncepcji języka deskryptorowego, uzyskała w pracy A. Kiszy najogólniejszą podstawę teoretyczną rozwoju systemów programowania, w postaci zamkniętego systemu dedukcyjnego, mogącego wszakże rozwijać się w różnych kierunkach [4]. Zrozumiałe jest więc zainteresowanie informatyków niemieckich, poszukujących dróg optymalizacji kodu prawnego i przejścia od systemu deskryptorowego, opartego na dysjunkcjach binarnych, do systemu kombinatorycznego, symulującego rzeczywistą dynamikę systemu prawa [5].

Sformalizowany opis statyki i dynamiki systemu prawa, jak każda tego rodzaju konstrukcja teoretyczna, wymaga wypracowania szeregu pojęć elementarnych. Wymienić tu należy koncepcję ilościowej cechy normy, która pozwala autorowi rozważać nie jakościowo pojęte normy prawa, lecz ich ilościowe przyporządkowania, tj. odpowiadające im liczby. Jest to koncepcja bardzo śmiała i płodna; w dotychczasowej literaturze przedmiotu spotyka się co najwyżej próby rozkładania oderwanych pojęć jakościowych na zmienne ilościowe [6].

Znany konserwatzm umysłu ludzkiego sprawia, że nowe i śmiałe koncepcje spotykają nierzadko więcej przeciwników niż zwolenników, toteż należy się spodziewać, że recenzowana teoria może zostać przez kogoś zaatakowana w tym właśnie punkcie, tym bardziej, że autor nie rozważa szerzej problematyki formalizacji systemu państwa i prawa. Reguł formalizacji uczy się na naszych uniwersytetach tylko stu-

---

*) Andrzej Kisza: Model cybernetyczny powstawania i działania prawa. Prace Wrocławskiego Towarzystwa Naukowego, Ossolineum Wrocław, 1970 str. 158, 42 ilustracje, 35 wzorów matematycznych, 3 tablice analityczne.

1) Por. Fr. Studnicki, Cybernetyka i prawo, Wiedza Powszechna 1969; M. G. Losano, Giuscibernetica, macchine e modelli cibernetici nel diritto, Torino 1969; obszerna recenzja J. Wróblewskiego, PAŃSTWO i PRAWO z 1970 r. na 12 s. 1021.

2) Por. Z. Rogoziński: Uwagi nad unowocześnieniem prawoznawstwa i nauczania prawa, ŻYCIE SZKOŁY WYŻSZEJ, 1974 r. nr 1 zwłaszcza str. 33—38; Cybernetyczne metody zarządzania, mat. z Sympozjum W-wa 22-26.04.1974 r.

3) Prace nad podstawami teoretycznymi programowania i automatycznego projektowania systemów, ograniczającymi element subiektywizmu związany z heurystyką, prowadzone są w całym świecie, lecz nie wyszły jeszcze dotychczas z fazy oderwanych, luźnych koncepcji teoretycznych, o zróżnicowanym poziomie ścisłości naukowej. Por. W. Marek, Z Pawlak: Information storage and retrieval system — mathematical foundations, PRACE C.O. PAN 1974; J. Kulikowski; Problemy optymalizacji struktury funkcjonalnej systemu informatycznego, Mat. II Sympozjum „System, modelowanie, sterowanie" PTC Zakopane 25.05.—31.06.74 r.

4) W oparciu o teorię zawartą w recenzowanej pracy, autor przygotowuje obecnie do druku probabilistyczną teorię dowodów procesu sądowego.

5) Myśl tę rozwija m.in. D. Suhr w artykule: Von DISUM zu ARSCOM, Vom Subscmionhilfeprogramm DISUM zum Konzept einer programmierten ars combinatoria, Arbeitspapiere Rechtsinformatik, zesz. 4 s. 7, J. Schweitzer Verlag, 1971. Dynamika systemu prawa wyraża się w funkcji zmiany znaczenia jednej normy prawa w konsekwencji zmiany innej normy przez organ prawotwórstwa.

6) Koncepcję taką referuje prof. Uniwersytetu w Heidelbergu, A. Podlech w pracy: Wertungen und Werte im Recht, Archiv des Öffentlichen Rechts 197 z. 2.

dentów matematyki, stosują zaś formalizację coraz częściej ekonomiści, socjologowie, psychologowie i prawnicy: stąd możliwość niedoceniania tych reguł przez jednych, a przeceniania ich przez drugich.

Jeżeli autor zakłada w recenzowanej pracy matematyczną strukturę przyrody i społeczeństwa, co nie byłoby u nas poglądem odosobnionym, a doświadczenie uczy, że sfera poznania jakościowego cofa się bez przerwy przed ofensywą modeli i metod ilościowych, uzasadniony jest, moim zdaniem, pogląd, że normom prawnym odpowiadają w rzeczywistości społecznej relacje o charakterze ilościowym, jakkolwiek kwantyfikacja ich na danym etapie rozwoju nauki może rodzić nieprzezwyciężalne trudności. Te trudności jednak, to właściwości podmiotu poznającego, ograniczające jego zdolność budowania modeli ilościowych, które należy odróżnić od właściwości samego przedmiotu. To założenie metamodelowe zwalnia — moim zdaniem — autora od obowiązku udowodnienia, że norma prawna posiada cechę liczbową, spełniającą akcjomaty arytmetyki.

Autor przejmuje klasyczną, wienerowską koncepcję cybernetyki, jako teorii samoregulacji systemu na podstawie sprzężeń zwrotnych za pomocą informacji o charakterze liczbowym, co pozwala na pomiar ilości informacji i wyrażenie jej w kodzie binarnym. Odcina się tym samym milcząco od takiego uprawiania cybernetyki, które zaciera różnicę między cybernetyką a semantyką języka naturalnego [7]).

Wobec trudności ustalenia ilości wejść i wyjść systemu społecznego dokonuje autor ich podjęciowej agregacji, wprowadzając koncepcję sprzężeń silnych i słabych, która wedle informacji autora, pochodzi od wrocławskiego matematyka, prof. dr B. Knastera.

Dynamikę skuteczności norm prawnych bada autor za pomocą skonstruowanego przez siebie współczynnika sprzężenia o charakterze kwantytatywnym. Współczynnikiem sprzężenia między oknami łańcucha cybernetycznego O' i O'', dotyczącym tej samej cechy nrmy $\bar{N}$, jest liczba

$$K_{O''}^{O'} = \frac{\bar{N}''}{\bar{N}'}$$

z której wyprowadza autor współczynnik sprzężenia zwrotnego:

$$K_{O'}^{O''} = \frac{1}{K_{O''}^{O'}}$$

Pojęcie regulacji prawnej pojmuje autor trojako:

a) tradycyjnie — jako regulację społeczeństwa prawem
b) cybernetycznie — jako regulację prawa sprzężeniami zwrotnymi
c) w sensie międzynarodowym — jako regulację nadukładową.

Szczególnie pasjonująca jest cybernetyczna analiza stosunków międzynarodowych, tzn. zagadnienie systemowych uwarunkowań wojny i pokoju.
Autor nie neguje, że system prawa nie jest jedynym systemem regulacji społecznej. Traktując jednak całokształt procesów regulacji społeczeństwa z punktu widzenia regulacji prawnej ujmuje on oddziaływanie innych systemów sterowania oraz ich reperkusje, bądź jako czynniki stymulujące wartość liczbową współczynnika sprzężenia zwrotnego, bądź jako znoszące się sygnały o znaku przeciwnym, wreszcie jako zakłócenia w funkcjonowaniu systemu prawa, które odróżnia od szumów własnych, wywoływanych przez nadmiar lub wadliwość norm prawnych.

Osią pracy jest model cybernetyczny powstawania i działania prawa, który składa się z łańcucha podstawowego *L, N, P, R* i *S* oraz wzmacniacza czyli regulatora prawnego: *J, A, E*. W modelu tym funkcjonalnie potraktowane społeczeństwo rozłożone jest na ogniwa *P, R, S*, stosownie do rozkładu skutków norm prawnych w czasie, czyli inaczej mówiąc, stosownie do długości czasu przetworzenia informacji niesionych przez normy prawne (w ogniwie *S* odkładają się skutki prawa z okresu stu i więcej lat). Model ten, jak i poszczególne jego ogniwa rozbudowuje autor stosownie do prowadzonych rozważań *).

Omawiany model powstał z zastosowania pojęć cybernetyki do empirycznej wiedzy o prawie i jego funkcjonowaniu w społeczeństwie. Najciekawsze, że mimo takich założeń, przy całkowitym wyeliminowaniu założeń z zakresu filozofii prawa, model prowadzi do potwierdzenia zasadniczych tez teorii państwa i prawa, ukazując w nowym świetle znane instytucje prawne oraz potwierdzając dedukcyjnie pozytywny lub negatywny wyrok historii wobec tych instytucji, np.: konieczność promulgacji norm prawnych, wyższość ustroju demokratycznego nad autokratycznym oraz procesu kontradyktoryjnego nad inkwizycyjnym [8]), metodologiczny podział wykładni prawa itp.

Jak już wspomniałem na wstępie, z punktu widzenia informatyki wartość praktyczna recenzowanej książki wyraża się w tym, że stanowi ona podstawę metodologiczną do budowy modeli symulacyjnych i opartych na nich programów służących do uzyskiwania precyzyjnych diagnoz i prognoz prawno-politycznych, dotyczących np. dynamiki przestępczości, strategii organizacji przestępczych, dynamiki nastrojów ludności oraz prawdopodobieństwa zaburzeń społecznych itd.

Wywody autora stanowić mogą również punkt wyjścia do opracowania — po raz pierwszy w historii — naukowych metod usuwania nadmiaru prawa. Autor, odosobniony najprawdopodobniej w światowej literaturze prawniczej, stawia niedostrzegany dotychczas problem badania szumów własnych regulacji prawnej. Widoczny wzrost tych szumów w państwach współczesnych prowadzi do wydawania norm „przeciwszumowych", które powodują dalszy wzrost szumów. W ten sposób inflacja przepisów rodzi nihilizm prawny, co w dziedzinie prawa wyraża się narastającym kryzysem zablokowania informacyjnego. Bronią przeciwko temu jest informatyka.

Autor mówi, że „gdyby udało się stworzyć system prawny, z którego zostałyby zupełnie usunięte szumy, to ilość norm prawnych bardzoby się zmniejszyła" (s. 54). Jest to bardzo kusząca perspektywa.

Burzliwy rozwój informatyki w ostatnim dziesięcioleciu sprawia, że wiele z ostatnio wydanych pozycji edytorskich można już zaliczyć do przestarzałych. Z tym większą satysfakcją można stwierdzić, że książka A. Kiszy nie tylko nie traci, lecz zyskuje teraz na aktualności, zasługując w pełni na nowe wydanie.

Jan Szymański

---

7) Przez ignorowanie zasadniczej różnicy między informacją traktowaną jako statystyczna miara nieokreśloności, a informacją w znaczeniu semantyki języka naturalnego. „Na poziomie maszyny znajdowaliśmy się jeszcze w świecie cybernetyki, którą interesuje sygnał. Wprowadzając człowieka, przeszliśmy w świat sensu. Rozpoczął się proces oznaczania, ponieważ sygnał nie jest już szeregiem jednostek dyskretnych, dających się oszacować w bitach informacji, lecz jest pewną formą — oznacznikiem, która adresat-człowieka będzie musiał wypełniać znaczeniem". U. Eco: Pejzaż semiotyczny. PIW, 1972, s. 52.

*) Oznaczenia: L — organ emisji norm (legislatywa), N — system norm, P — psychiczno-czynnościowe skutki prawa, R — Bezpośrednie skutki prawa, S — wtórne lub odległe skutki prawa, J — sądownictwo (jurysdykacja), A — administracja, E — organy egzekucyjne.

8) Zasada kontradyktoryjności w procesie sądowym, która zwyciężyła w całym świecie cywilizowanym, z wyjątkiem krajów dyktatury policyjnej, zakłada autonomizację oskarżenia, obrony i sądzenia, które to funkcje w procesie typu inkwizycyjnego teoretycznie pełni jedna osoba sędziego.

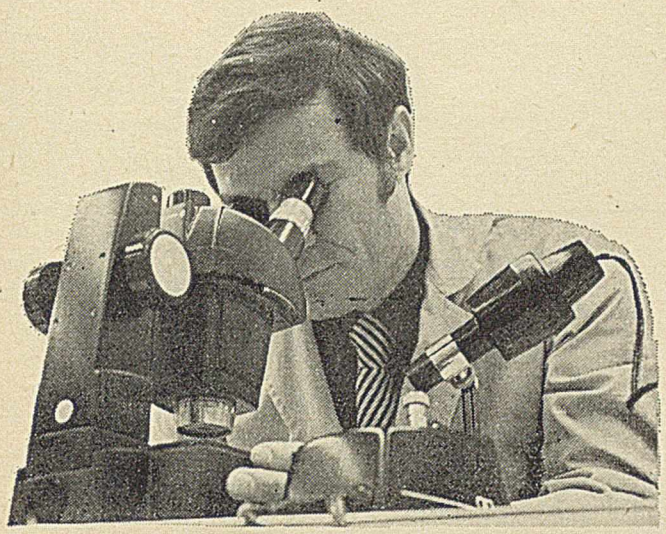
Łatwo było mówić, że jesteśmy najlepsi, gdy — pierwsi i jedyni w świecie — zaczęliśmy produkować komputerowe taśmy magnetyczne.

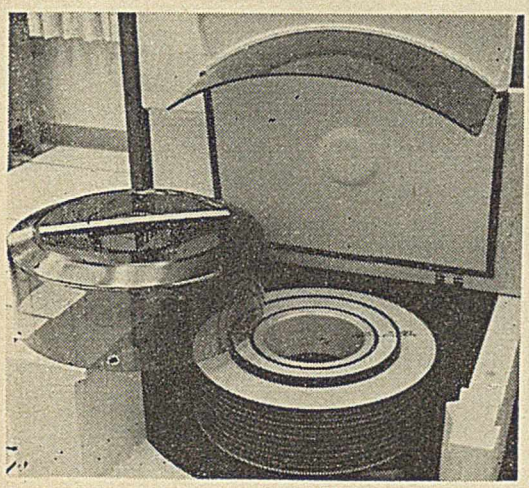
To było 20 lat temu, w 1953 r. specjalnie powleczona taśma acetatowa naszej produkcji zapoczątkowała nową erę elektronicznego przetwarzania danych. Z chwilą ukazania się naszych taśm na rynku producenci komputerów mogli zaprojektować urządzenia do magazynowania danych z nieosiągalną dotąd gęstością zapisu informacji i szybkością ich odszukiwania. 3 M prowadząc intensywne prace badawczo-rozwojowe z myślą o potrzebach przemysłu komputerowego — stale udoskonalało swoje wyroby, ulepszając technologię, zwłaszcza powłok magnetycznych i materiałów podkładowych. Jako pierwsi wprowadzaliśmy kolejno na rynek:

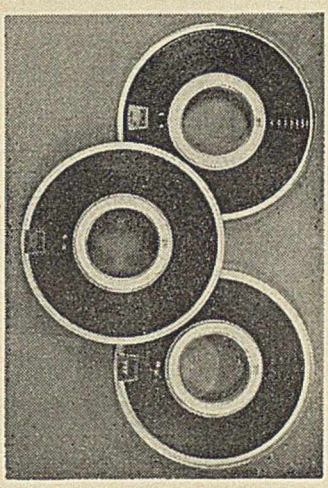
- taśmę na podkładzie poliestrowym
- taśmę z 50 mikrocalową warstwą ochraniającą powłokę tlenkową
- taśmę przystosowaną do trudnych warunków eksploatacyjnych
- taśmę o przedłużonym okresie eksploatacji i dużej odporności na uszkodzenia powłoki tlenkowej („dropout")
- taśmę testowaną wyłącznie aparaturą elektroniczną
- taśmę eliminującą uszkodzenia powodowane nieprawidłową eksploatacją.

Taśmy ze znakiem „Scotch' stały się podstawą ustanowienia międzynarodowych norm dla taśm magnetycznych i — tradycyjnie już — spełniają one wymagania stawiane przez różnorodne konstrukcje sprzętu komputerowego. Produkujemy w oparciu o samodzielnie wytwarzane tlenki, materiały podkładowe oraz własne receptury spoiw.

Nośniki magnetyczne produkujemy w nowocześnie wyposażonych fabrykach, gdzie sprawdzone kwalifikacje personelu technicznego i surowa kontrola produkcji stanowią o najwyższej jakości naszych wyrobów.
Chcemy być nie tylko pierwsi, ale i najlepsi!
A oto nasze najnowsze wyroby:



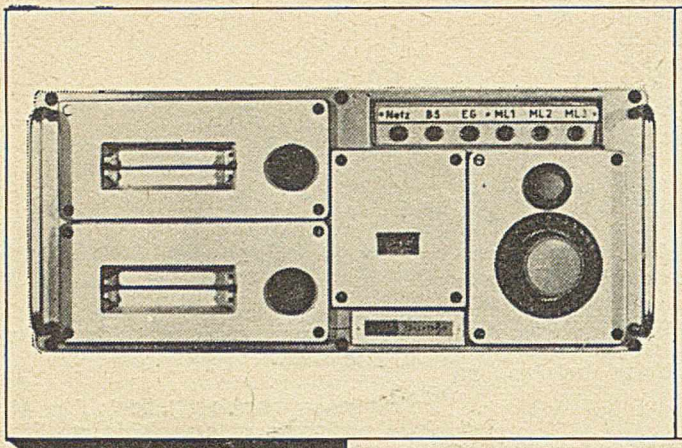


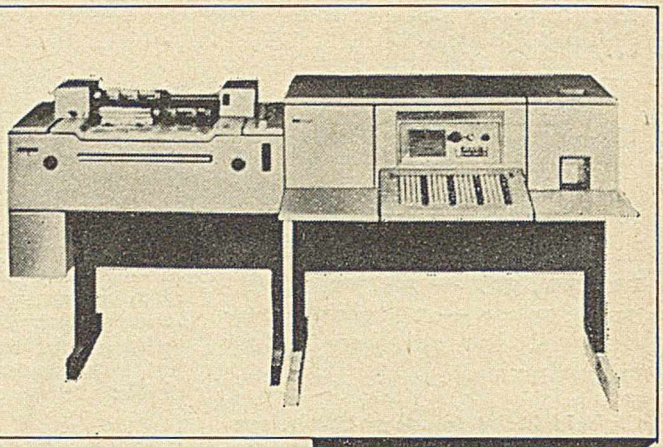


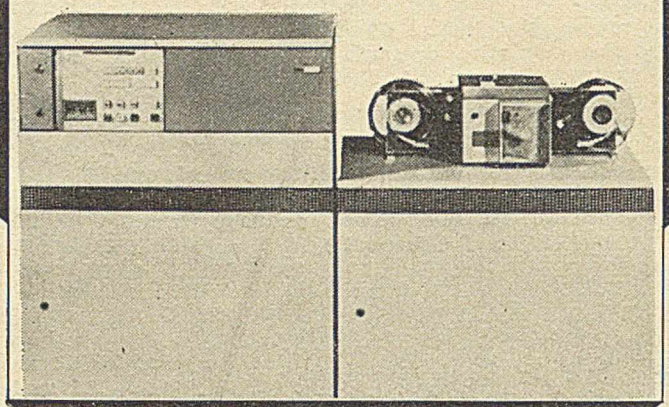
Komputerowa taśma magnetyczna 3M „Scotch" marki # 700 charakteryzuje się teksturowanym podłożem zabezpieczającym przed zarysowaniem, zaklesz-czaniem oraz zanieczyszczaniem powierzchni.

Pakiet dyskowy odporny na uszkodzenia („crashgurad") charakteryzuje się unikalnymi własnościami powłok, która chroni przed uszkodzeniami powierzchni magnetycznej wskutek zatarcia głowicą („headerash"), a w przypadku poważniejszych zatarć, zmniejsza tworzenie się aglomeratów tlenku oraz powstawanie zanieczyszczeń głowic zapisu/odczytu.

3M (EAST) AG
CH 6301 ZUG BAARERSTRASSE 8 SWITZERLAND
TELEPHONE 042 23 21 31 TELEX 78 873 CABLE TRIMINCO ZUG







daro 1600

Jeżeli dane mają być ewidencjonowane z wysoką techniczną efektywnością zastosowania, to tylko: daro 1600.

daro 1600 oznacza półautomatyczny system ewidencjonowania danych, odznaczający się wieloma możliwościami.

Jego części składowe: zdecentralizowane zespoły wyszukiwania i sterowania, stanowiska prezentowania danych, drukarki.

Sposób działania systemu: ręczne lub automatyczne ewidencjonowanie danych numerycznych lub alfanumerycznych w bezpośrednim sąsiedztwie ich powstawania. (Transmisja wprowadzonych informacji wprost do elektronicznego systemu przetwarzania danych w celu przetwarzania w czasie rzeczywistym lub partiowo). Drukarki zapewniają zarówno protokółowanie danych wprowadzonych, jak i wydruk danych wyprowadzonych z komputera.

daro 1600, ze swoimi rozmaitymi możliwościami konfiguracji stanowi wysokosprawną instalację drugiej peryferii.

Może być oczywiście sprzęgany bezpośrednio z wieloma elektronicznymi systemami przetwarzania danych.

Prosimy zwracać się o informacje.
Nasza dokumentacja jest do Waszej dyspozycji.
Przedstawicielstwo BME w Polsce
Biuro Techniczno-Handlowe przy Ambasadzie NRD
ul. Filtrowa 62 m. 63,
02-057 Warszawa
Sprzedaż i informacje:
PREDOM — org.
ul. Górskiego 9,
00-033 Warszawa

bme

WCT/15/K/74-B